

09/674864

PCT/US99/10002

WO 99/59078

1 / 47

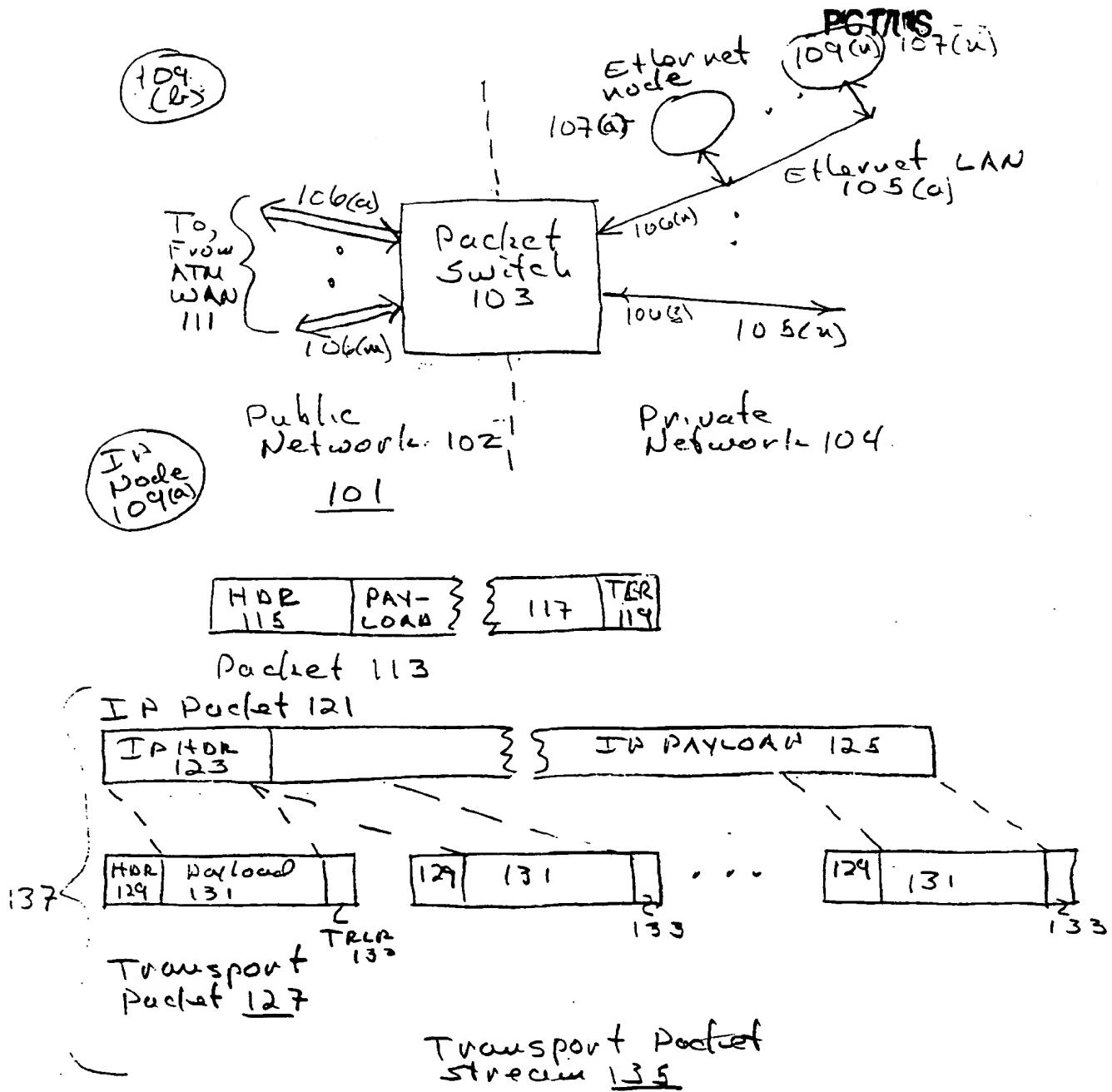


Fig. 1 Prior Art

09/674864

PCT/US99/10002

WO 99/59078

2 / 47

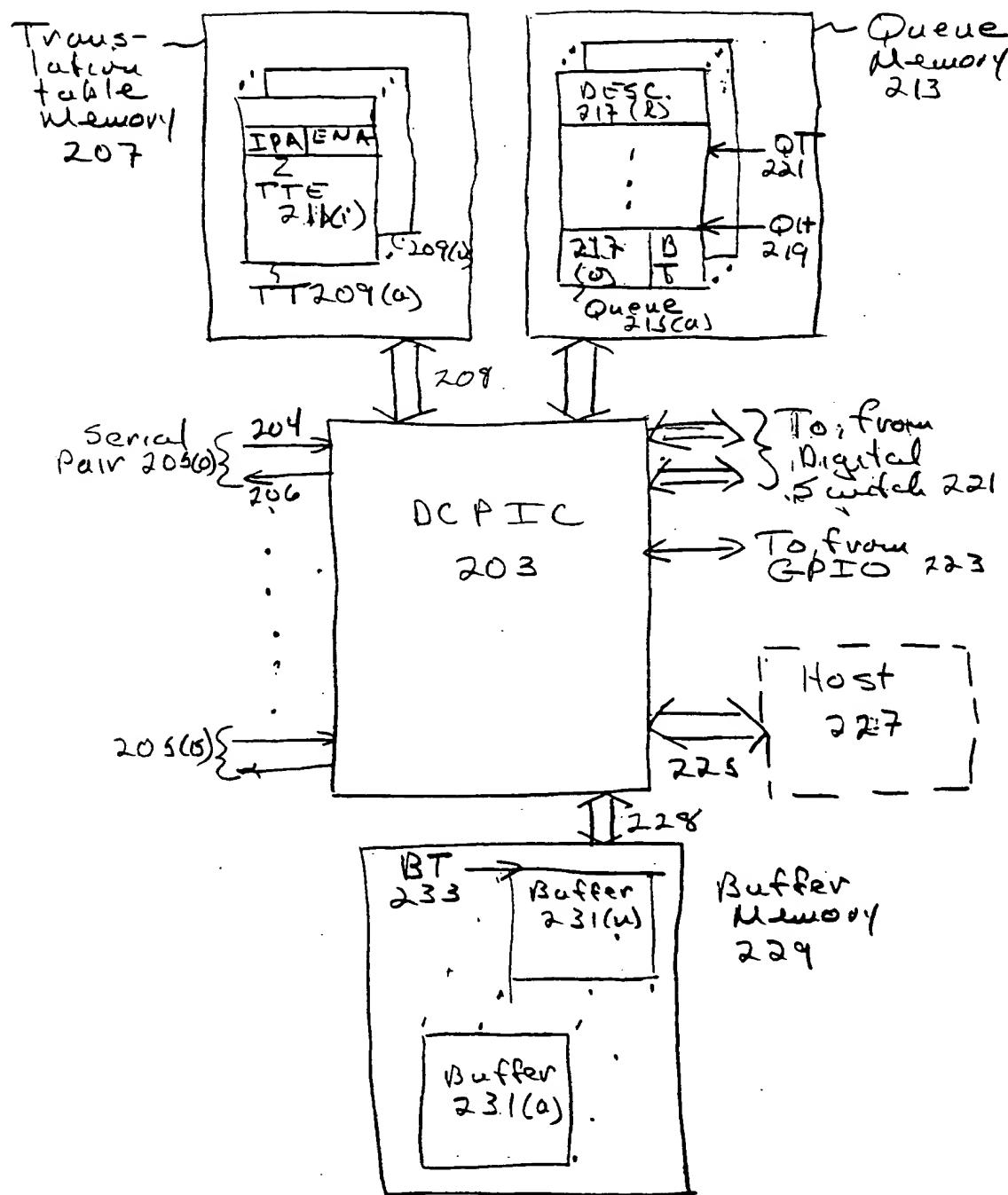
201

Fig. 2

09/674864

PCT/US99/10002

WO 99/59078

3 / 47

10 CHAPTER 1: INTRODUCING THE DIGITAL COMMUNICATIONS PROCESSOR

Company Confidential  
Preliminary Draft

Figure 1 - DCPA1 Block Diagram

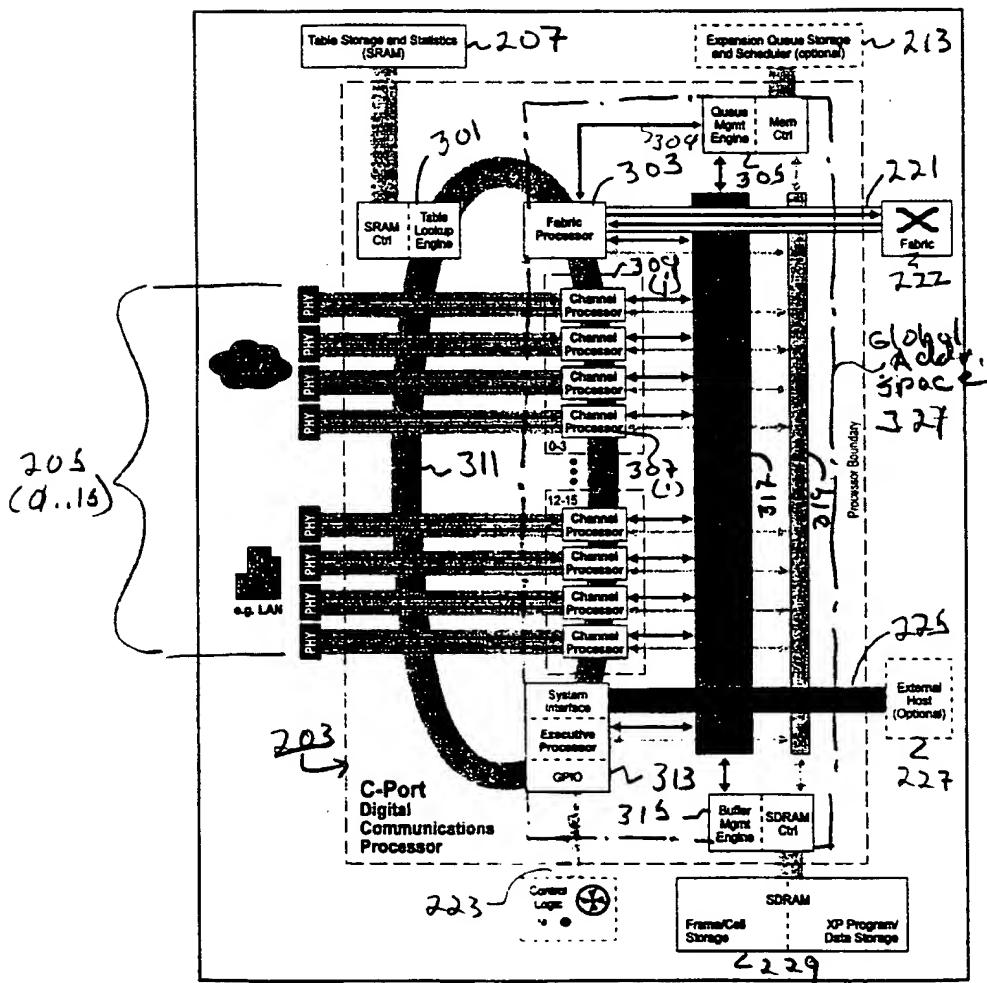


Fig. 3

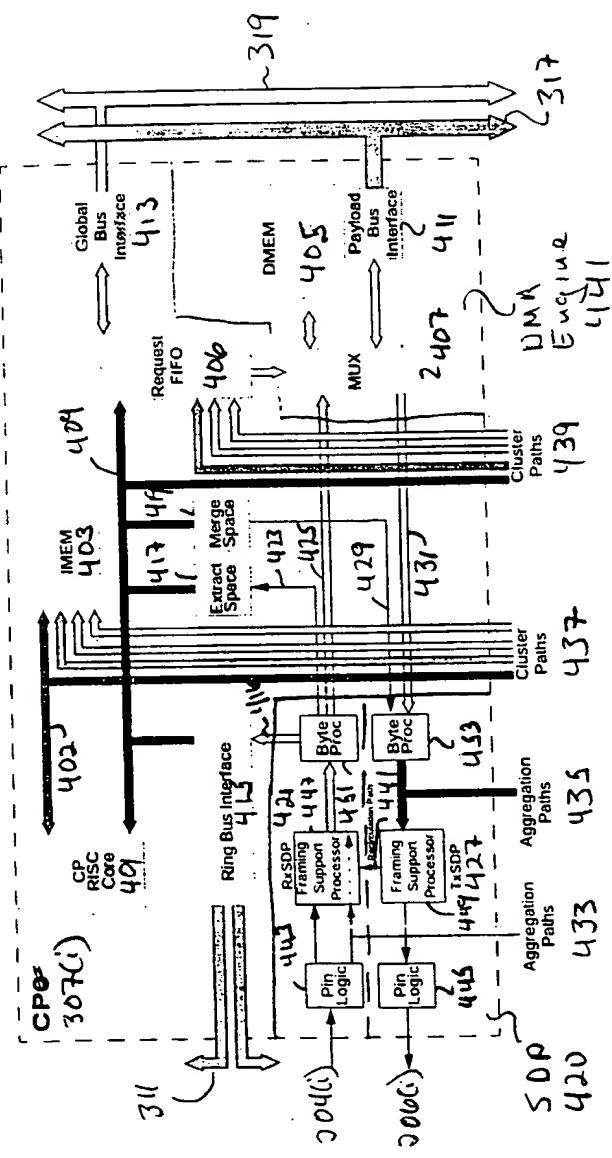


Fig. 4

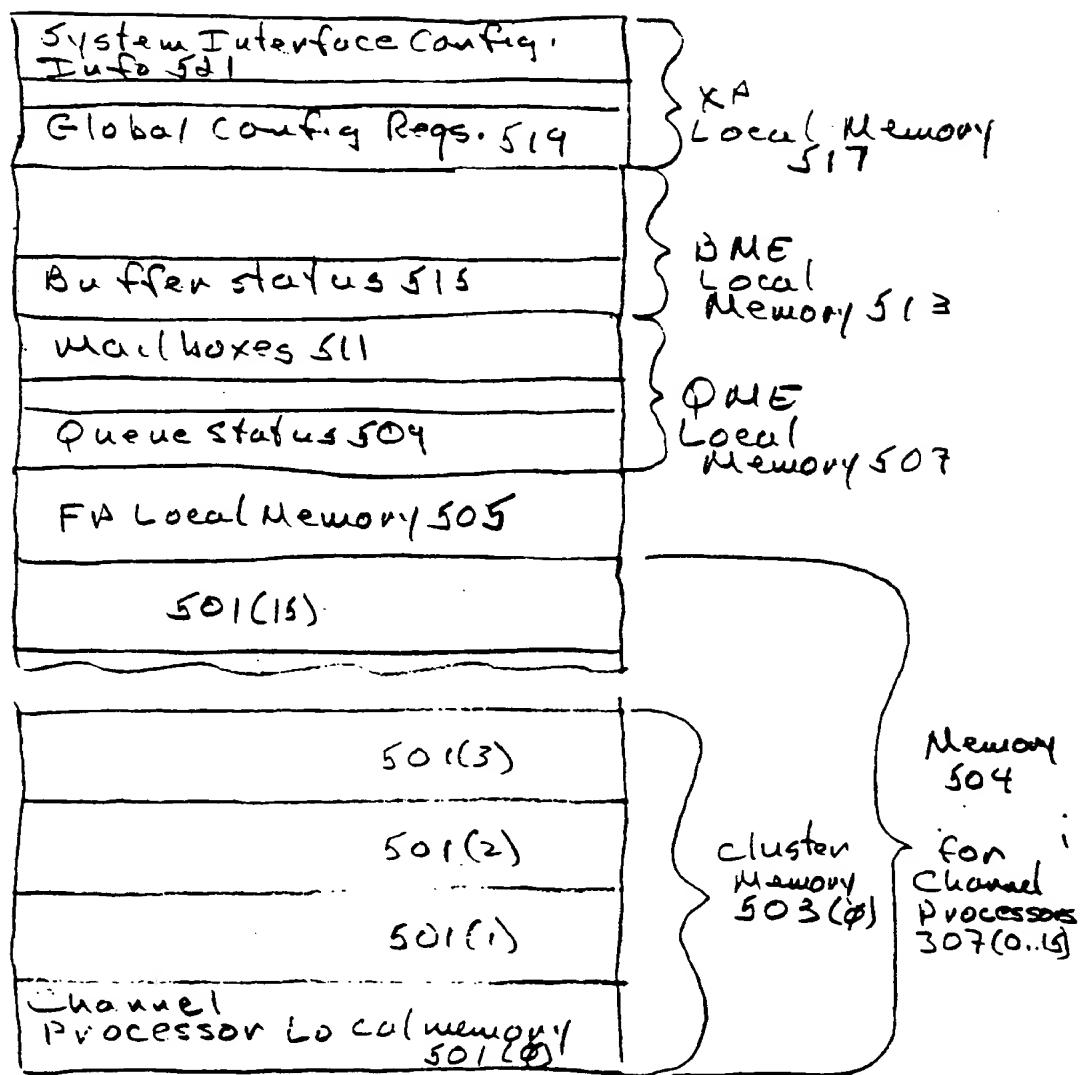


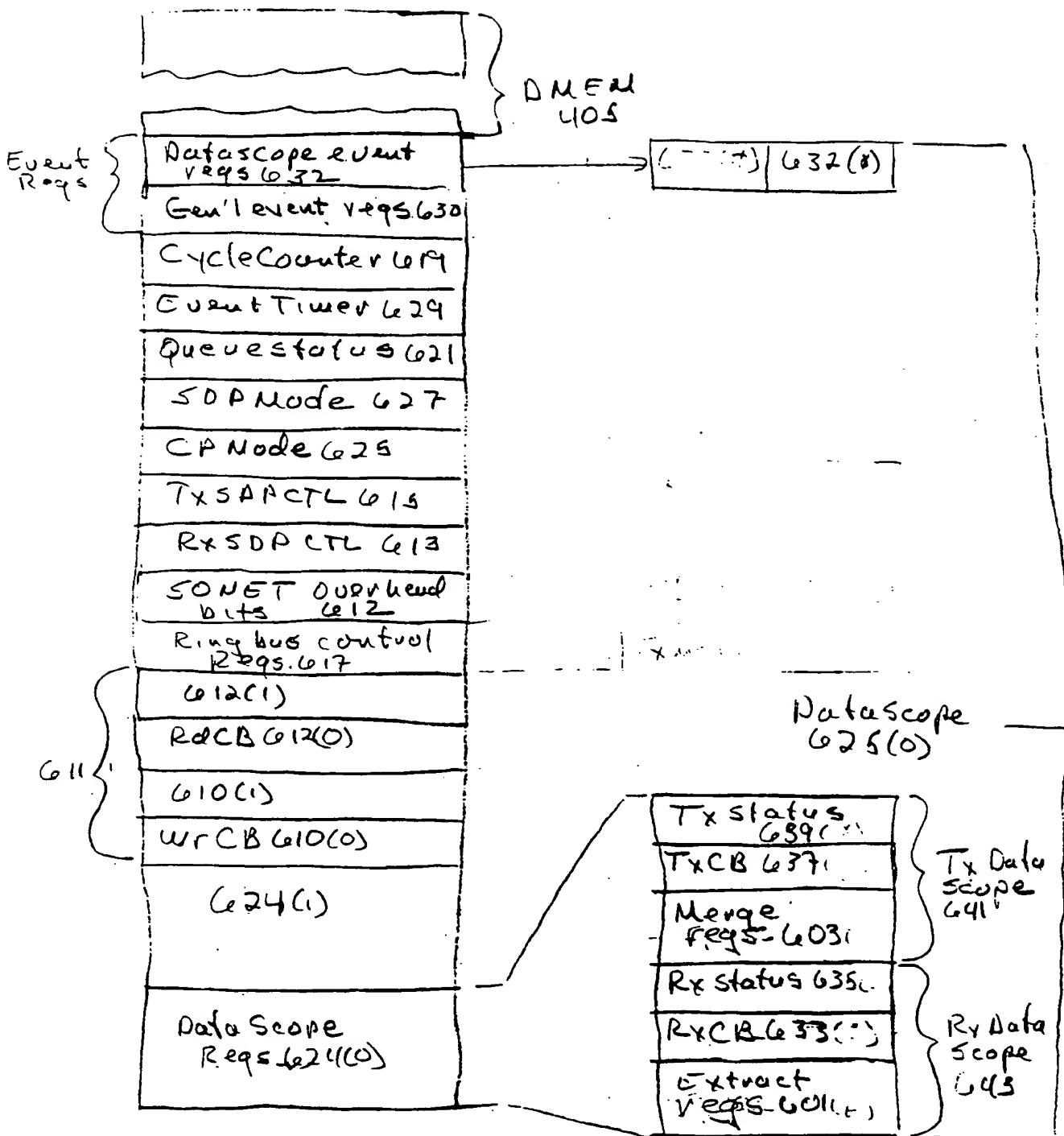
Fig. 5

09/674864

PCT/US99/10002

WO 99/59078

6 / 47



(1(i))

= 19.C

09/07/1984

PCT/US99/10002

WO 99/59078

7 / 47

Company Confidential  
Preliminary Draft

Data flow in the DCP - 15

0962464 - 033001

Figure 2 - DCP-1 Receive Thread of Execution Flowchart

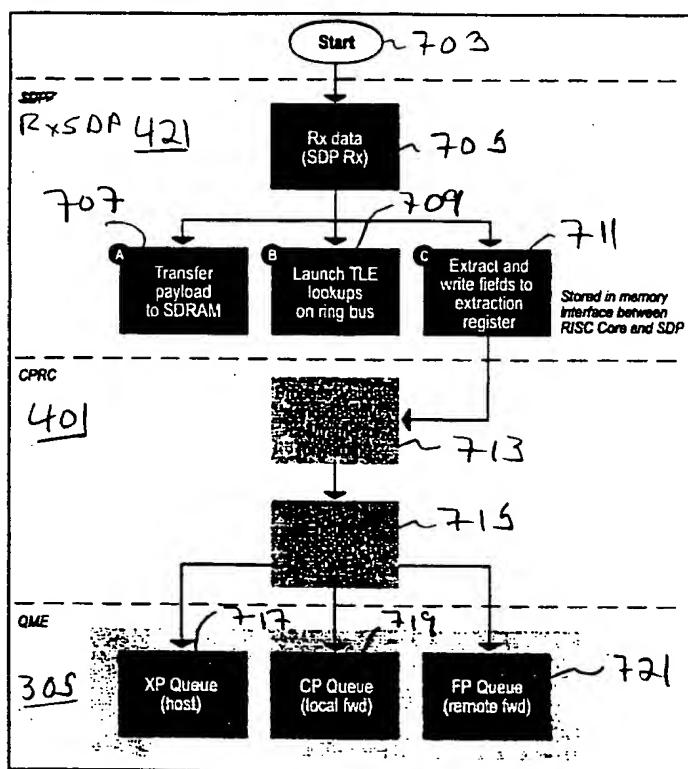


Fig 7

09/674864

Figure 8 DCP-1 transmit Execution Thread Flowchart

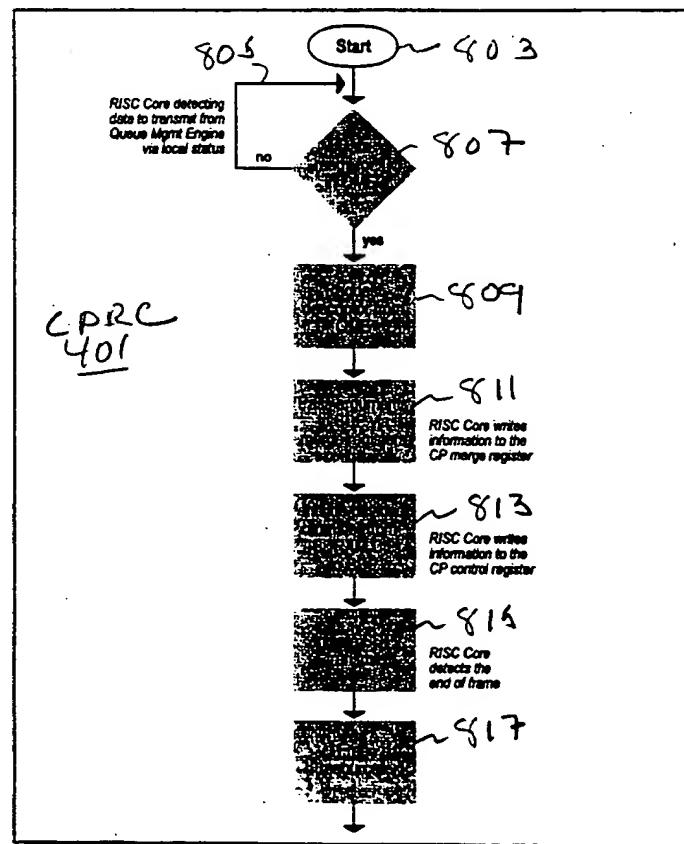


Fig. 8

09/674864

PCT/US99/10002

WO 99/59078

9 / 47

OWN 935	L5:Lφ 937	Busy 941
------------	--------------	-------------

635

BTAG 933			
OFFSET 931			
Av 929	NR 927	EVN 925	OWN 921
SOPST 915	EOP 927	BCTL ST 919	
Length 911			
Buffer Pool No 909			
DMAEM OUT ADDR 907			
Tx Reg Addr. 905			
Rx Reg Addr. 903			
DMAEM Byte Addr 901			

633

{ RXCBCTL 913}

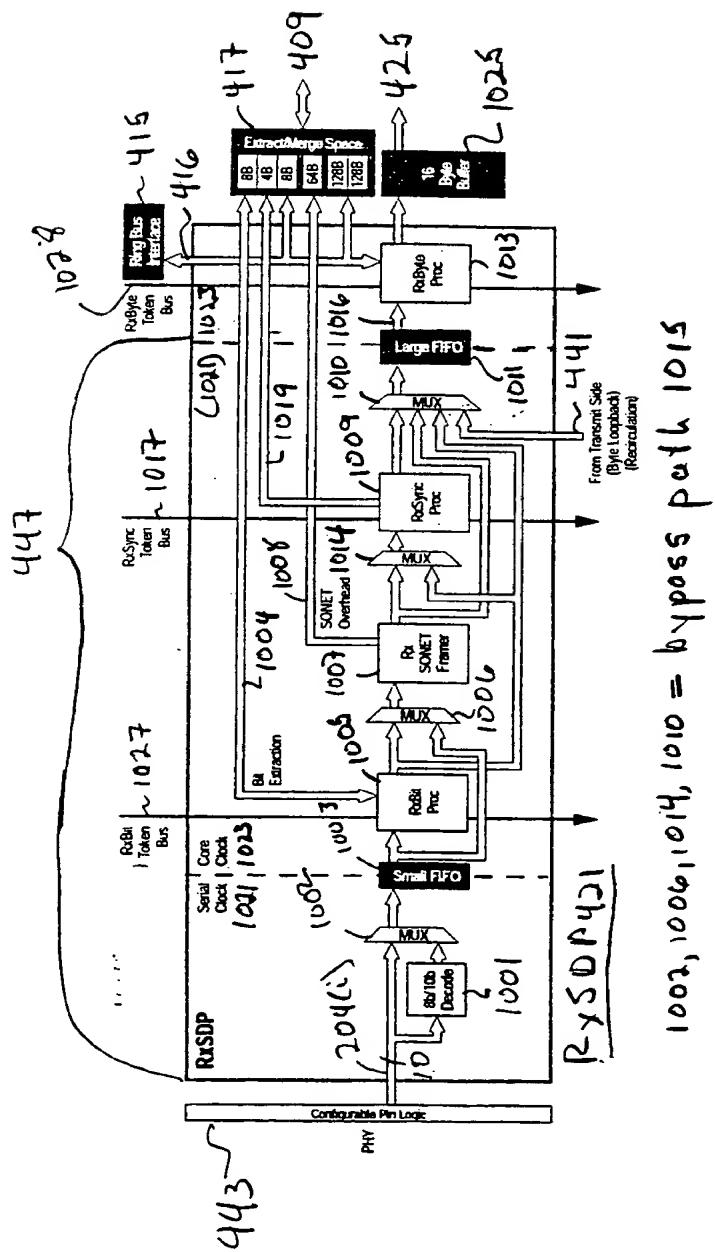
Fig. 9

09/674864

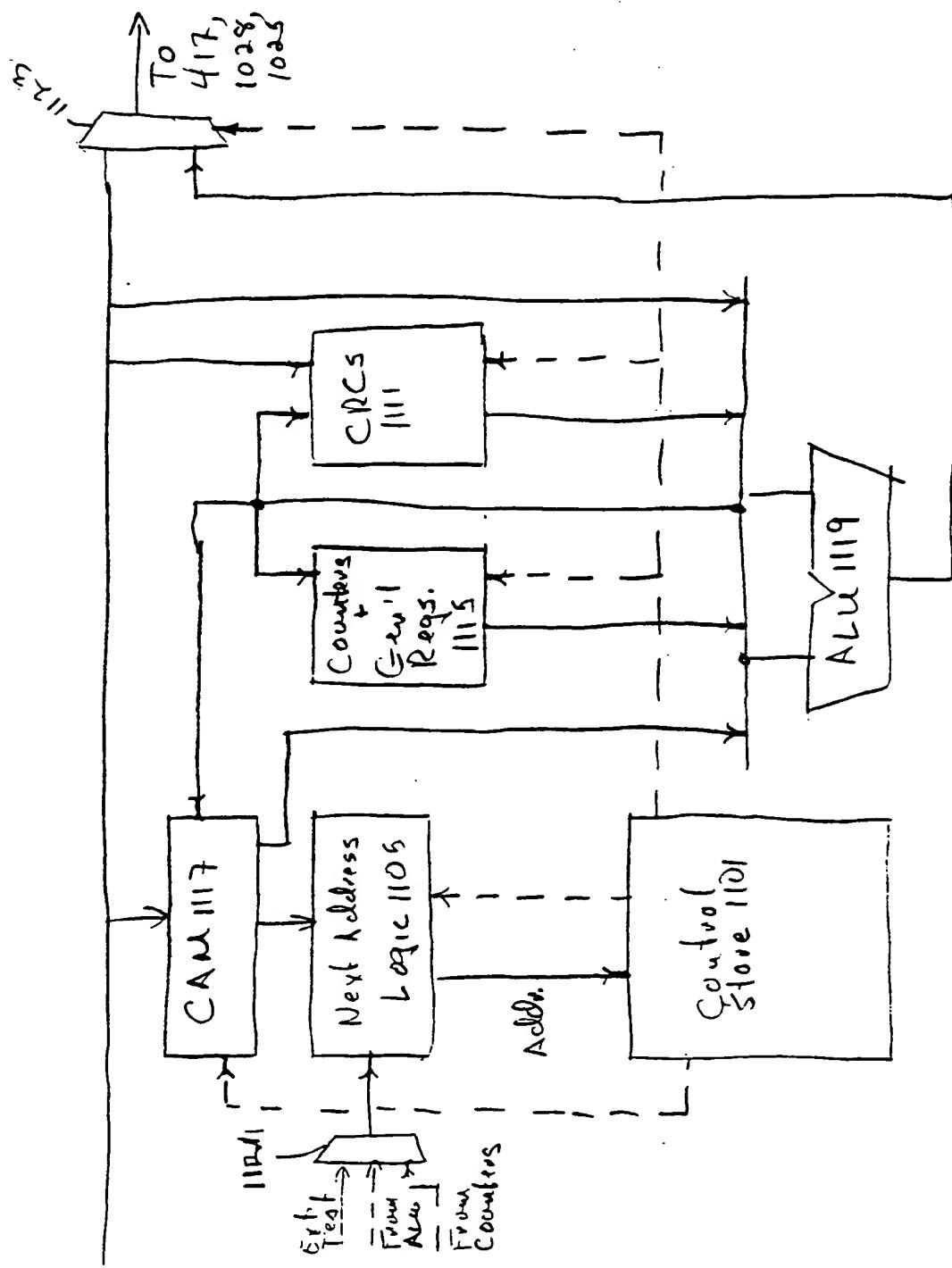
PCT/US99/10002

WO 99/59078

10 / 47



09/07/864



1013

Fig. 11

09/674864

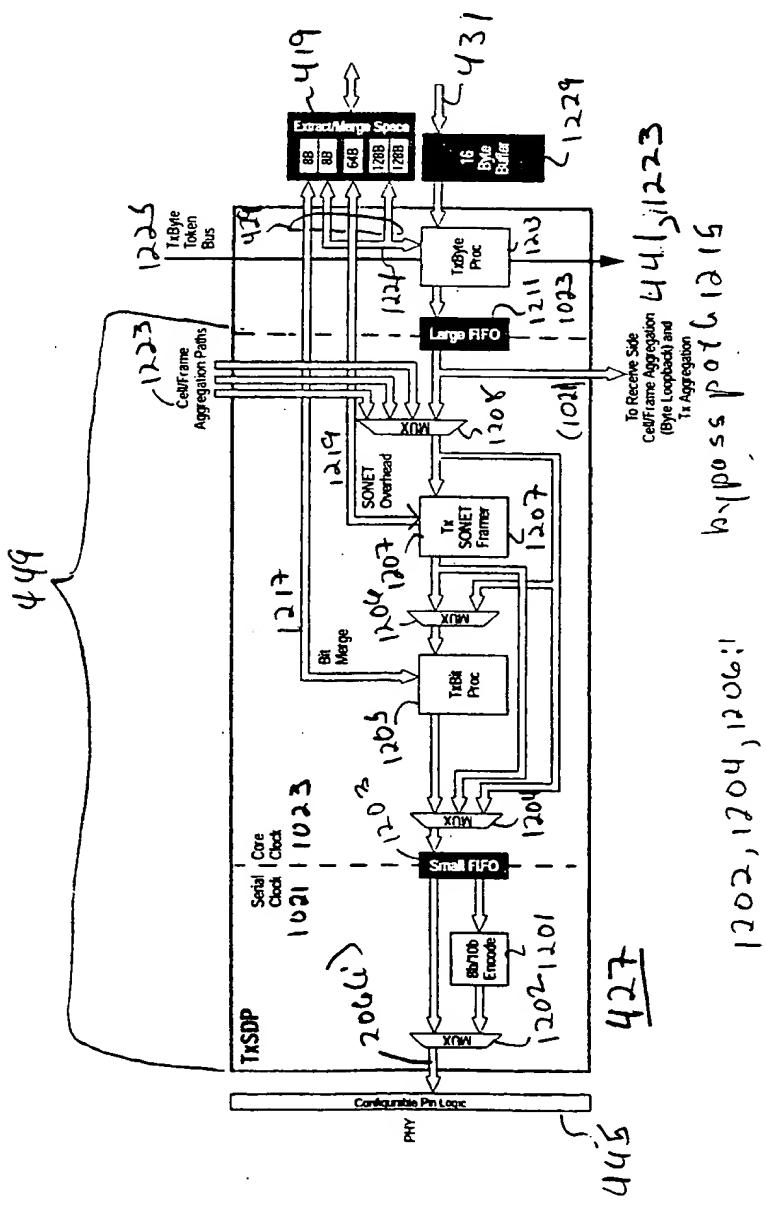


Fig. 12

09/674864

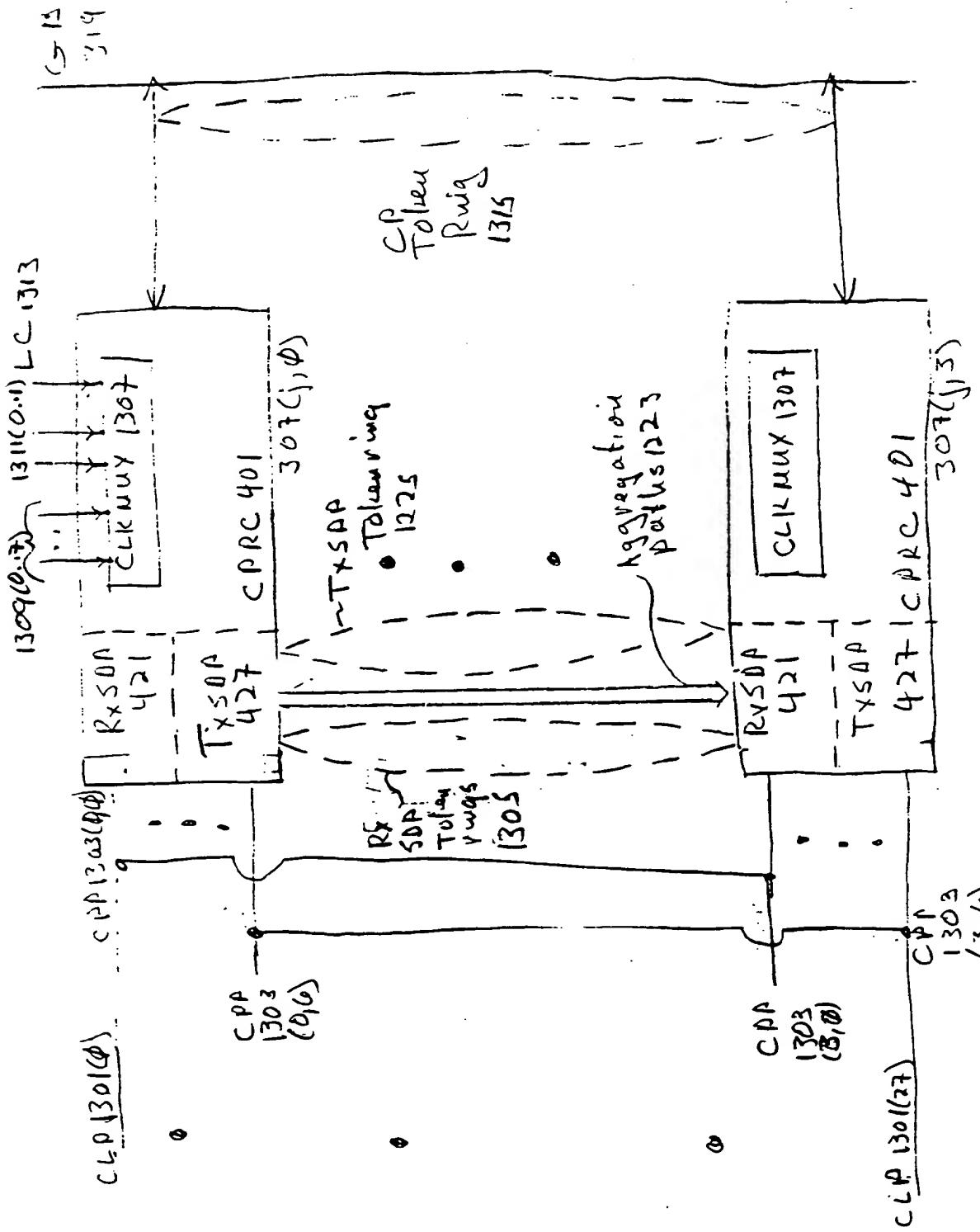


Fig. 13

09/674864

PCT/US99/10002

WO 99/59078

14 / 47

Company Confidential  
Preliminary Draft

CP Instruction Memory 29

**CP Instruction Memory**

Each channel shares access to a 16kB IMEM among a cluster of four adjacent CPs as shown in Figure 7. The IMEM is configured as four sub-arrays, with each CP in the cluster given access to the arrays, one per cycle, in fixed round-robin order. With this simple interleaved scheme, the four adjacent RCs can access this memory at nearly full bandwidth.

When adjacent channels are configured to handle similar communication protocols, the large shared memory can contain both channel-specific code and cluster-shared code such as exception routines.

At initialization time, the 16kB array can be divided so that each CP gets a dedicated 4kB sub-array. This array allocation removes all CP contention for IMEM (but also removes the opportunity to share code among CPs). The memory configuration options result in roughly the CPRC performance shown in Table 6-1 for non-blocking code. The *optimized* column means that the compiler has placed code such that the branch target address bits <3:2> equal the branch fall through address bits <3:2>. CPRC instruction references outside of the shared local memory space are not supported.

Table 3 Channel RISC Core Instruction Execution Efficiency

IPC	IPC optimized	CP IMEM configuration
.85	.90	4 CPs sharing 16kB
.95	N/A	each CP accessing a single 4kB sub-array

Figure 7 Local and Shared Memory in a Channel Cluster

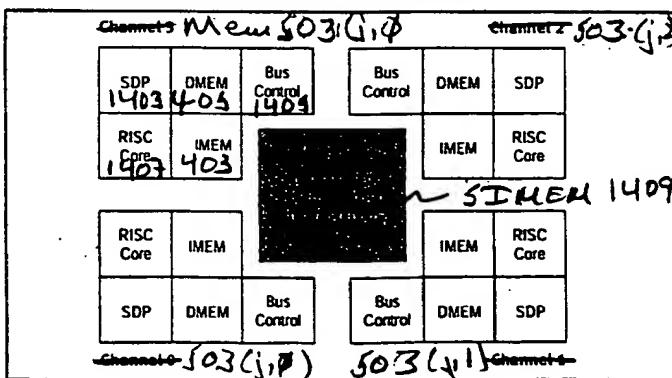


Fig. 14

509

09/674864

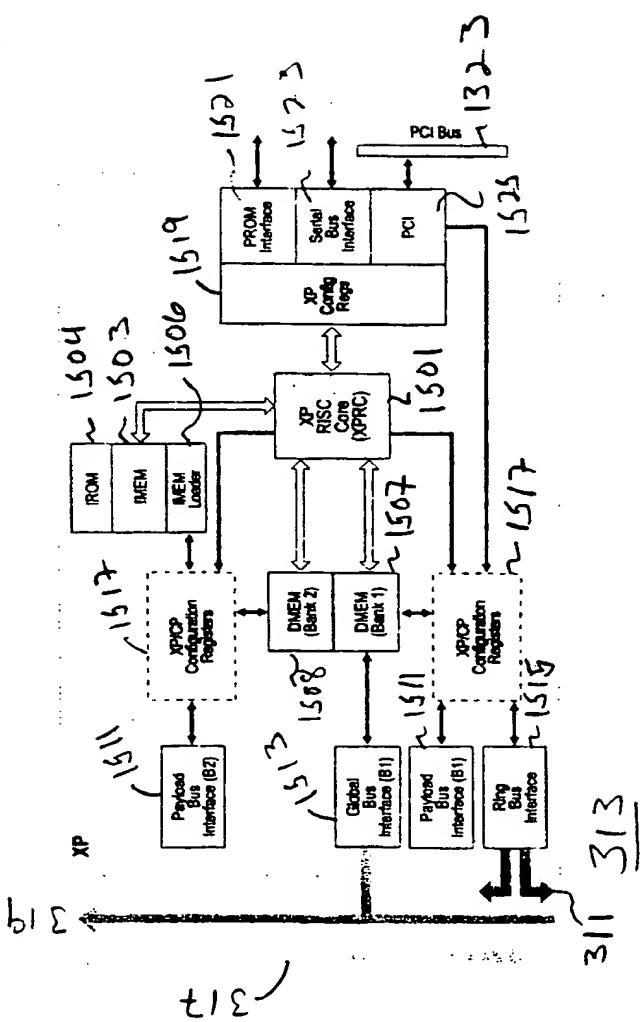


Fig. 15

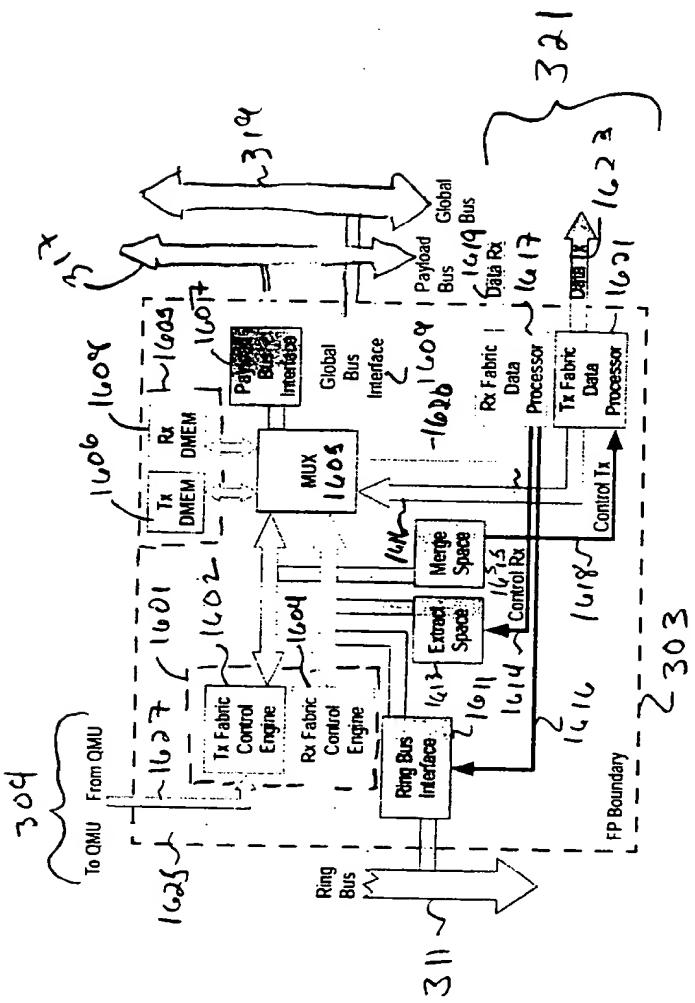


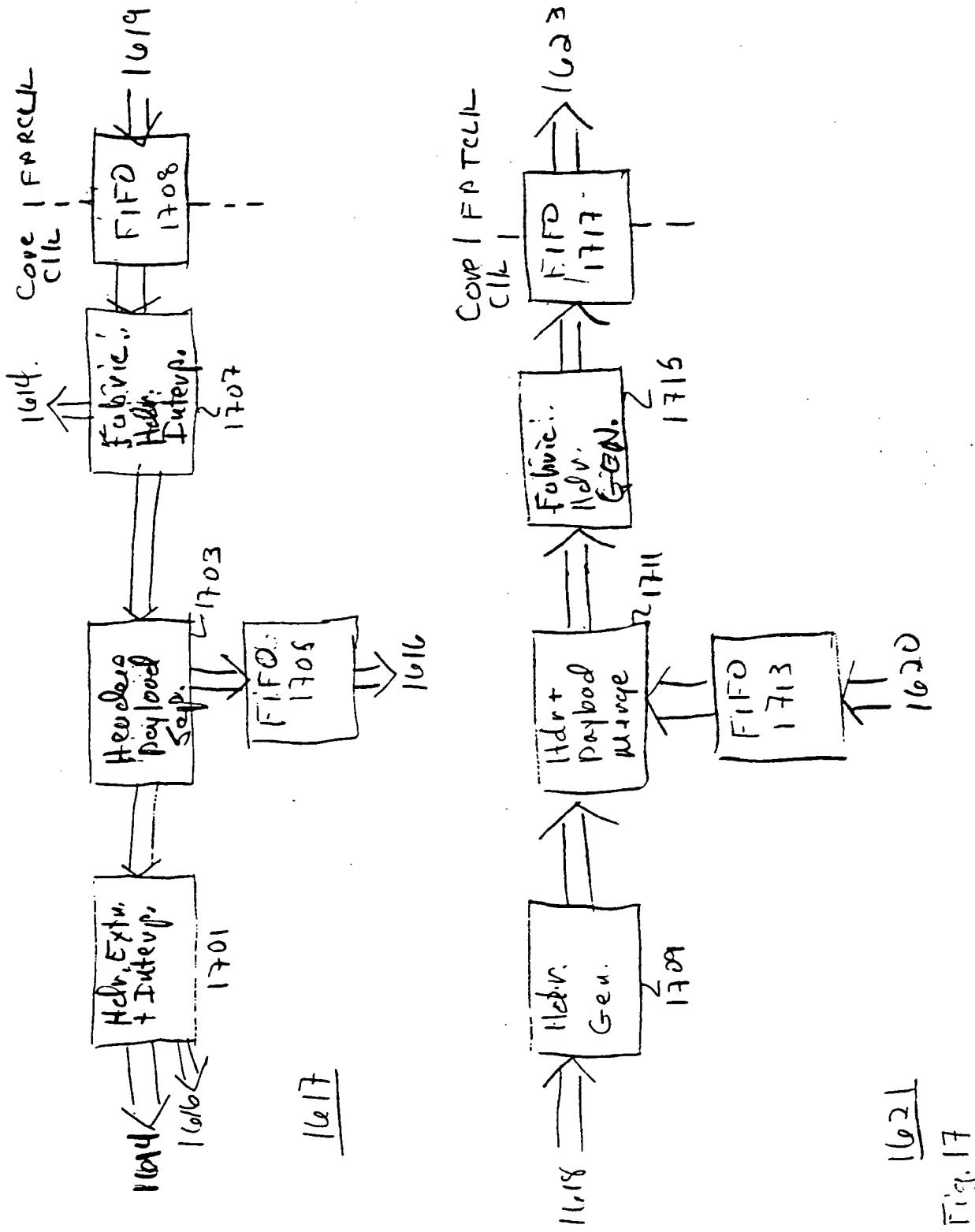
Fig. 16

09/674864

PCT/US99/10002

WO 99/59078

17 / 47



09/674864

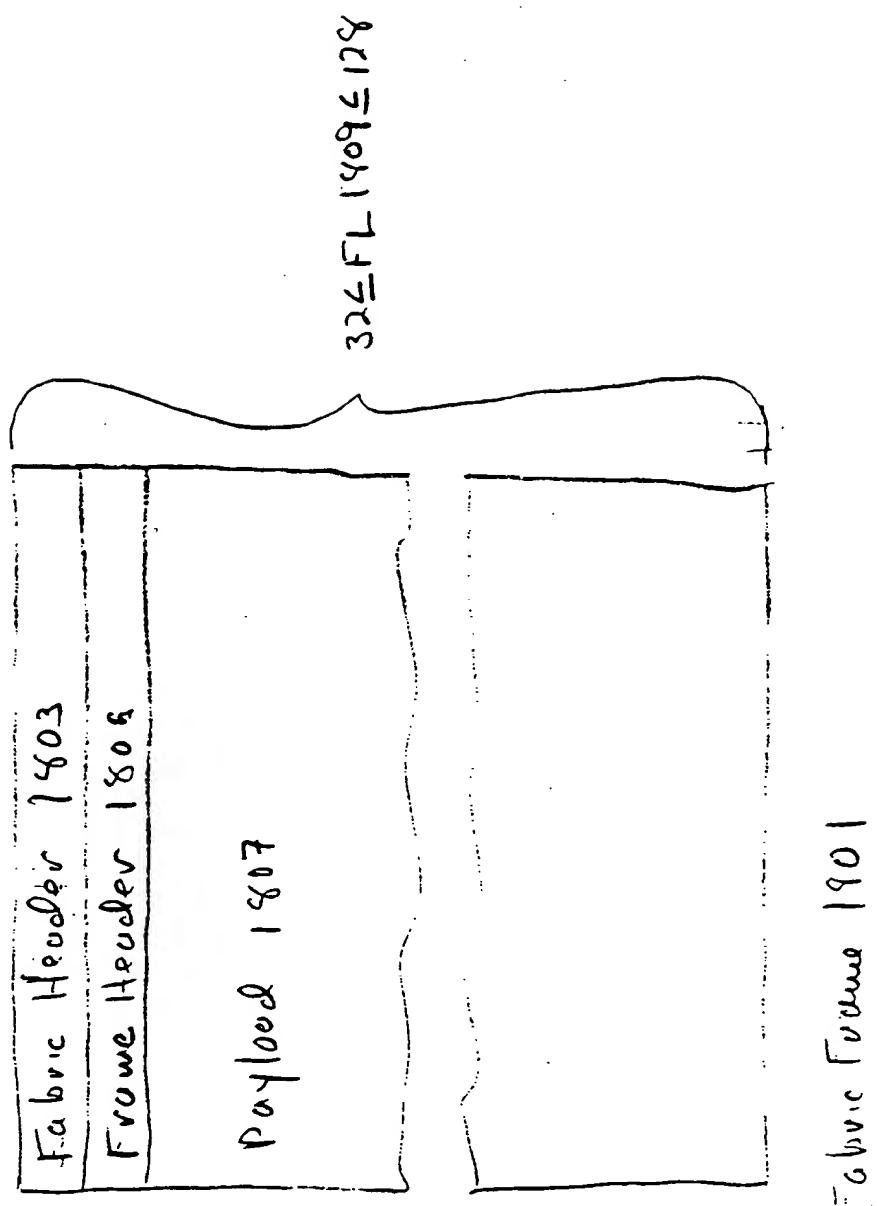
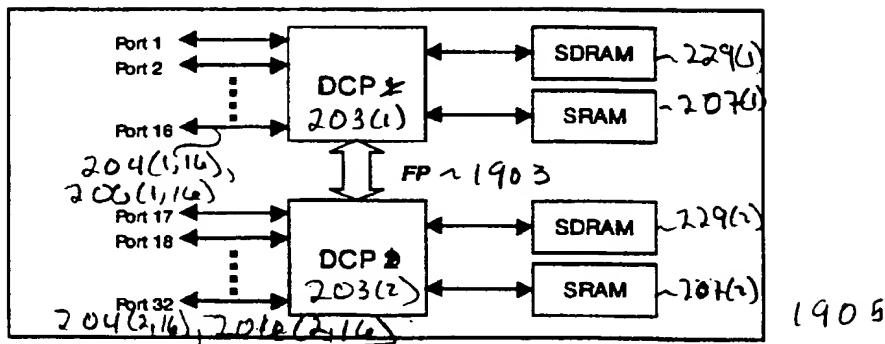


Fig. 18

09/674864

**Figure 12 Two DCP-1 Application**

When more than two DCP-1s are required in a system, a switching fabric is utilized. The switching solution has two or more FP-type ports and provides a mechanism for switching cell- or packet-based data from one DCP-1 to another. An homogeneous, multi-DCP-1 application is shown in Figure 13.

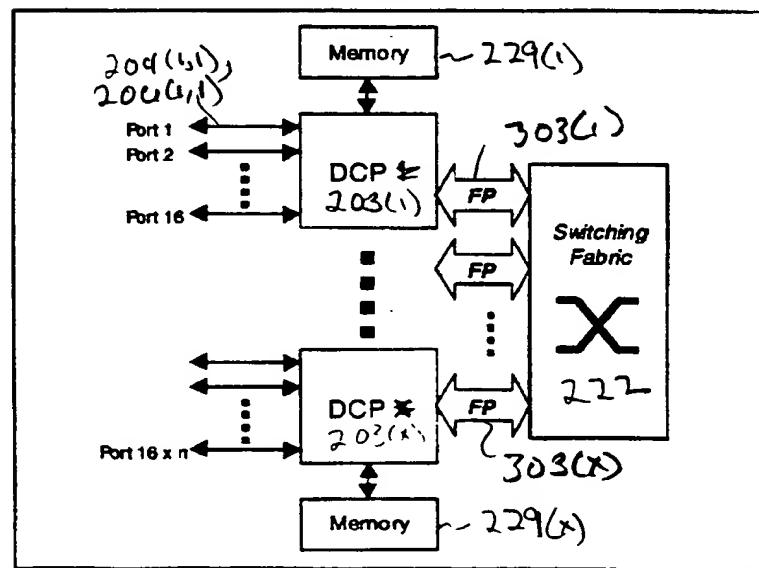
**Figure 13 Multiple DCPs with Switching Fabric**

Fig. 19

09/674864

Figure 14 Heterogeneous DCP-1 Switching Application

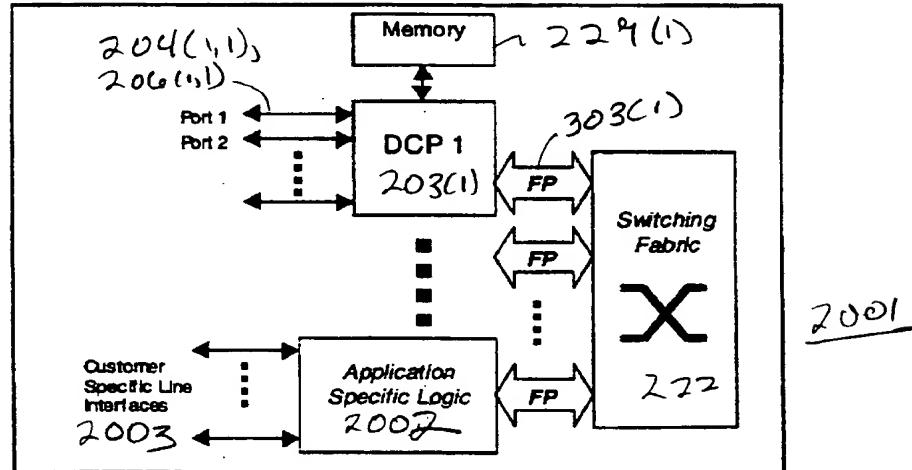
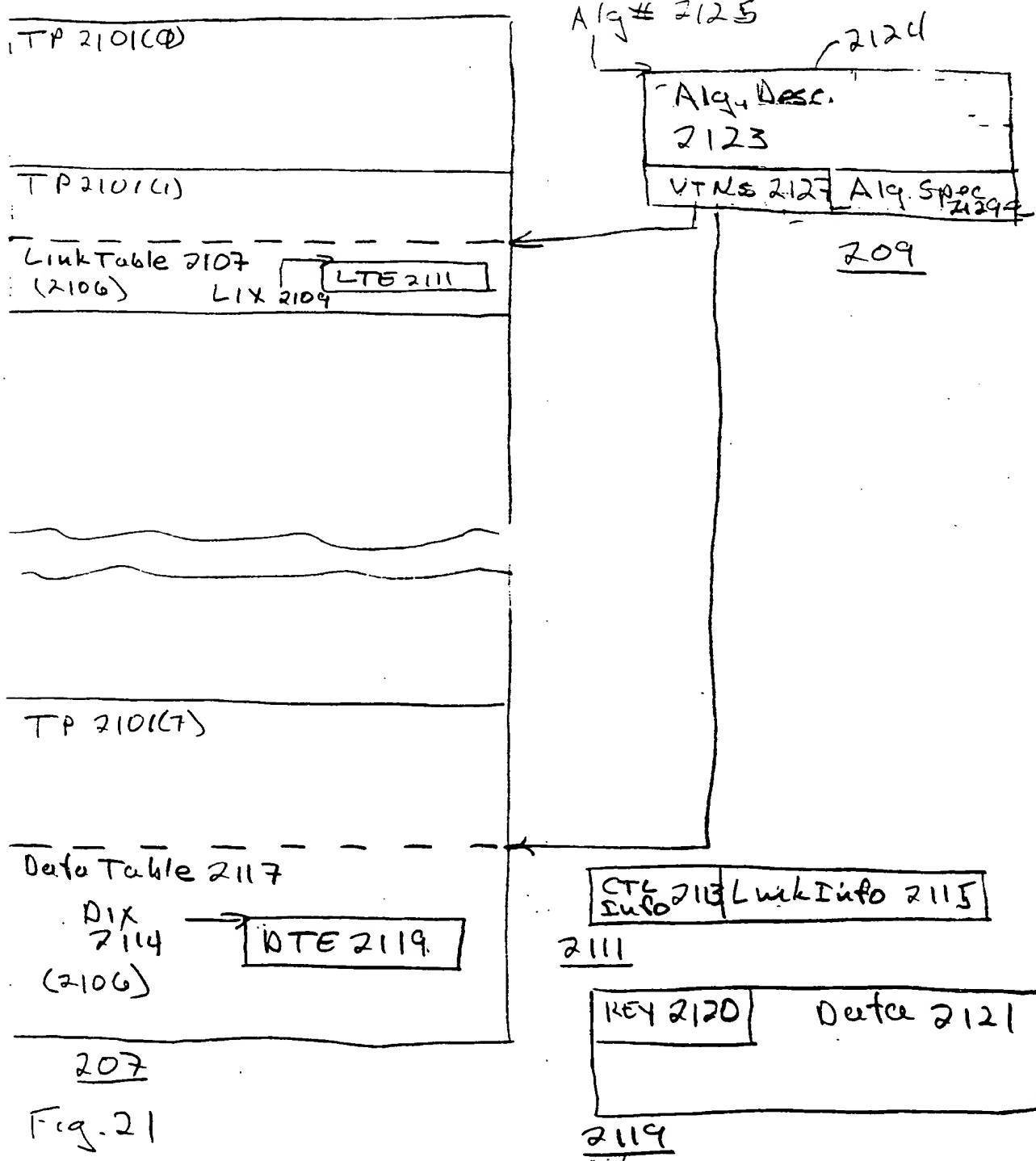


Fig. 20

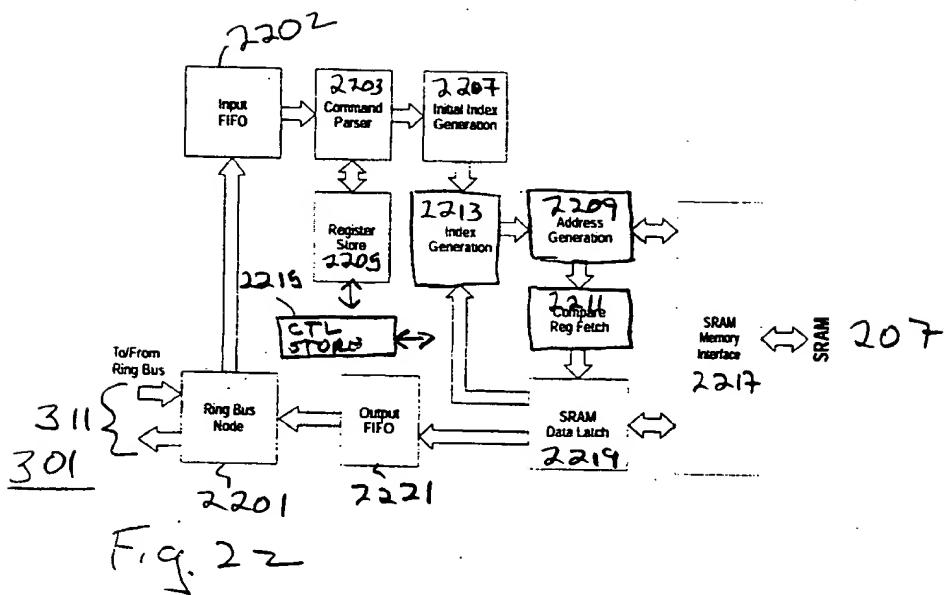
09/674864



09/674864  
PCT/US99/10002

WO 99/59078

22 / 47



09/674864

23 / 47

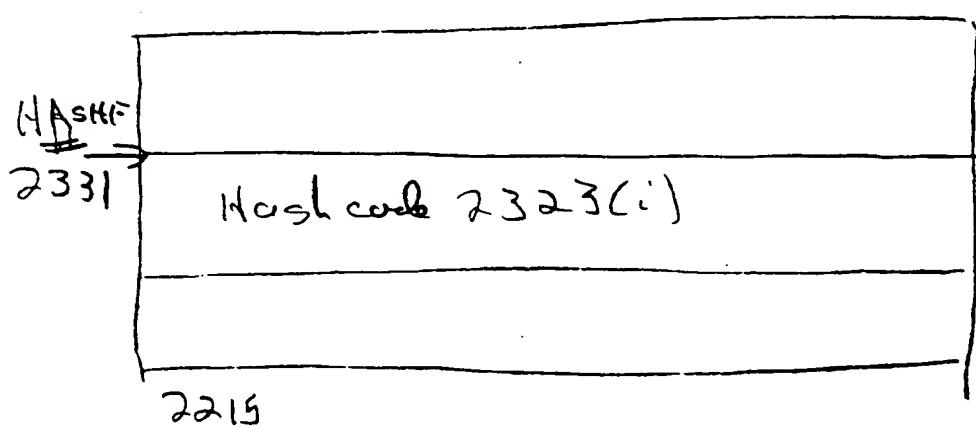
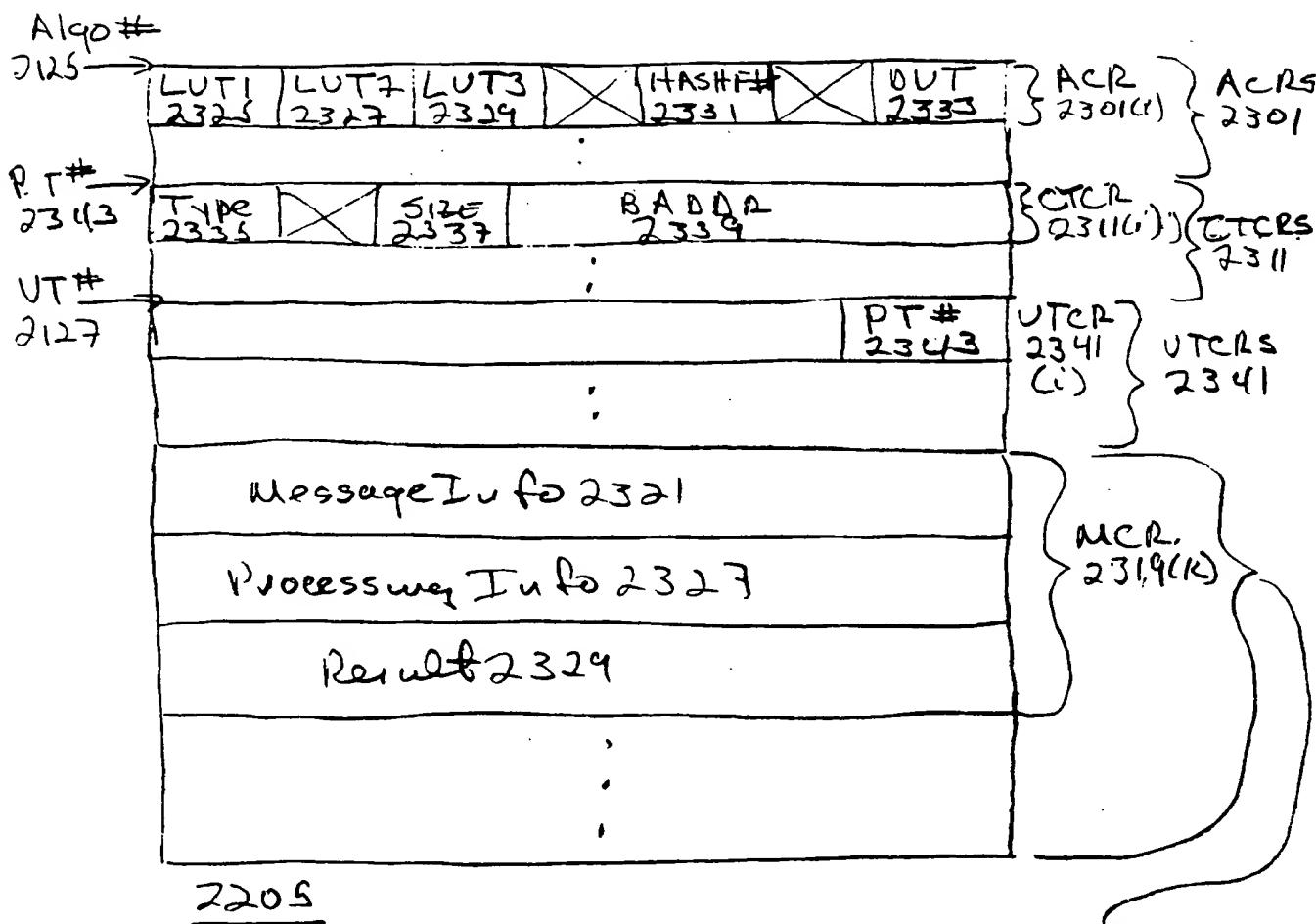


Fig. 23

09/674864

PCT/US99/10002

WO 99/59078

24 / 47

<b>Command</b>	<b>Command ID</b>	<b>Return Data</b>	<b>Description</b>	
Write(vtable#, index, mask, data, offset, length)	0x2	None	Write data into a virtual table at index.	~ 2401
Read(vtable#, index, offset, length)	0x3	Data	Reads data from a virtual table.	~ 2403
Find(alg#, key)	0x6	Physical Table, Index, Error	Finds a key using alg#. Sets Ring Bus Error Flag if key is not found.	~ 2405
FindW(alg#, key, data, offset, length)	0x4	Pass/Fail, Index, Error	Writes data into a table using a key. Sets Ring Bus Error Flag if the key is not found.	~ 2407
FindR(alg#, key, data, offset, length)	0x5	Pass/Fail, Index, data	Reads length dwords of data from a vtable# using a key at offset dwords. Sets Ring Bus Error Flag if the key is not found.	~ 2409
XOR(vtable#, Index, data/pcrc, offset, mask, crc, last)	0x1	None or CRC in CRC mode.	XORs up to a 32 bit value to offset. Only masks of up to four consecutive bytes are valid. A special mode exists for CRC calculations.	~ 2411
Add(vtable#, index, data, offset, mask)	0x7	None	Adds up to a 32-bit value to offset. Only masks of up to four consecutive bytes are valid.	~ 2413
WriteReg(reg_addr, data)	0x0,0x10	None	Write data to TLE register at reg_addr.	~ 2415
ReadReg(reg_addr, data)	0x0,0x11	Data	Read data from TLE register at reg_addr.	~ 2417
Echo(data)	0x0,0x04	Data	Returns data from TLE. For test purposes.	~ 2419
Nop()	0x0,0x05	None	Inserts a NOP into the TLE pipe.	~ 2420

Fig.24

09/674864

PCT/US99/10002

WO 99/59078

25 / 47

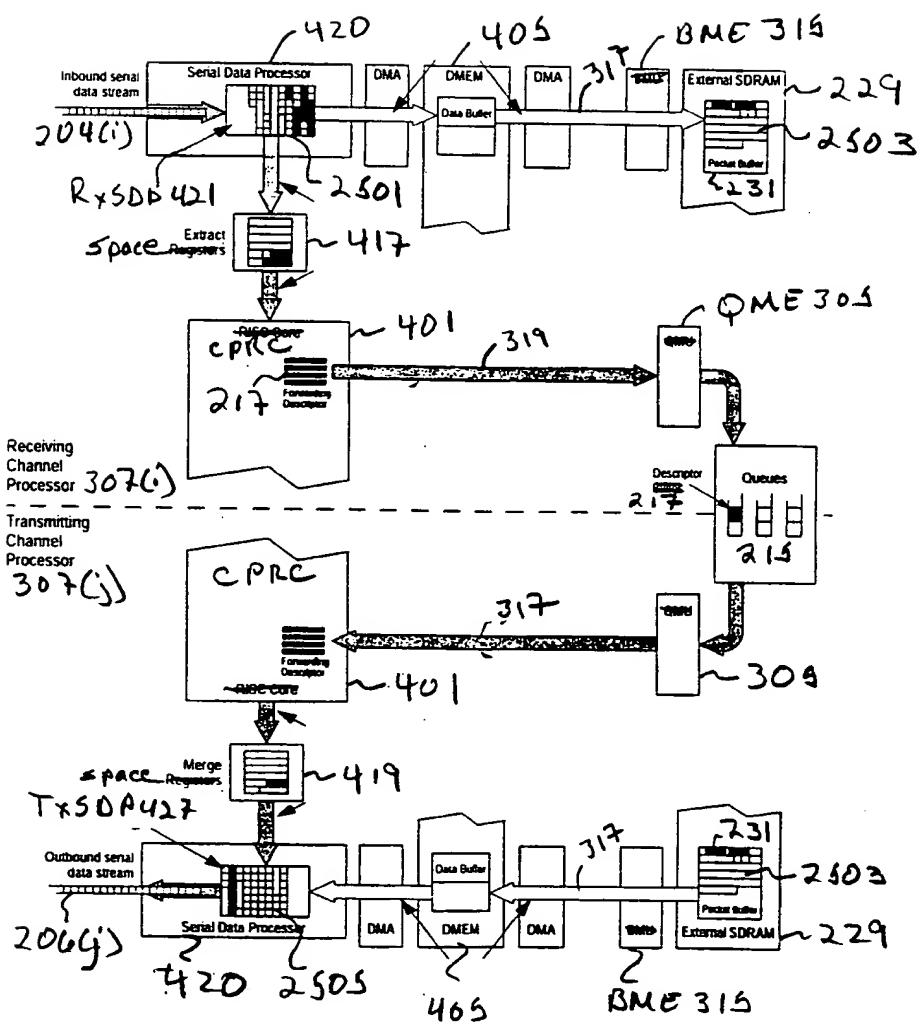


Fig. 25

09/674864 - 0330031

26 / 47

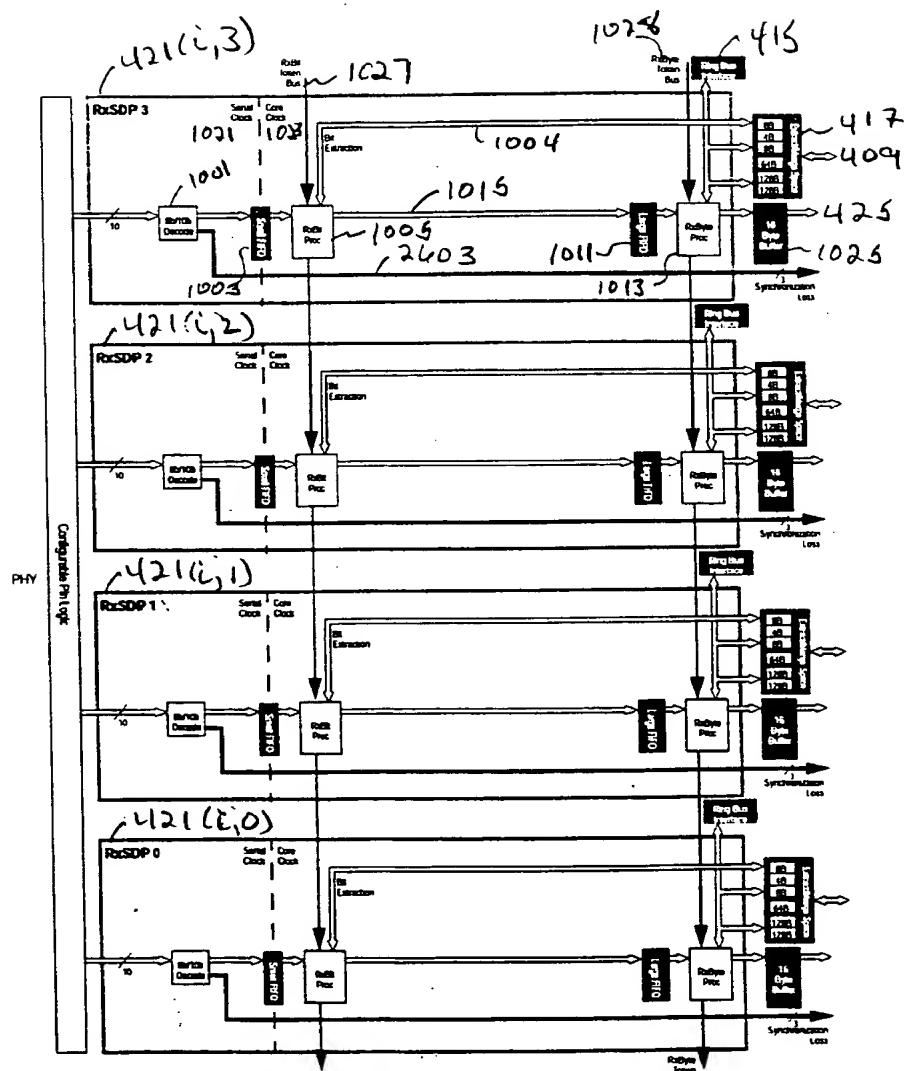


Fig. 26

09/674864

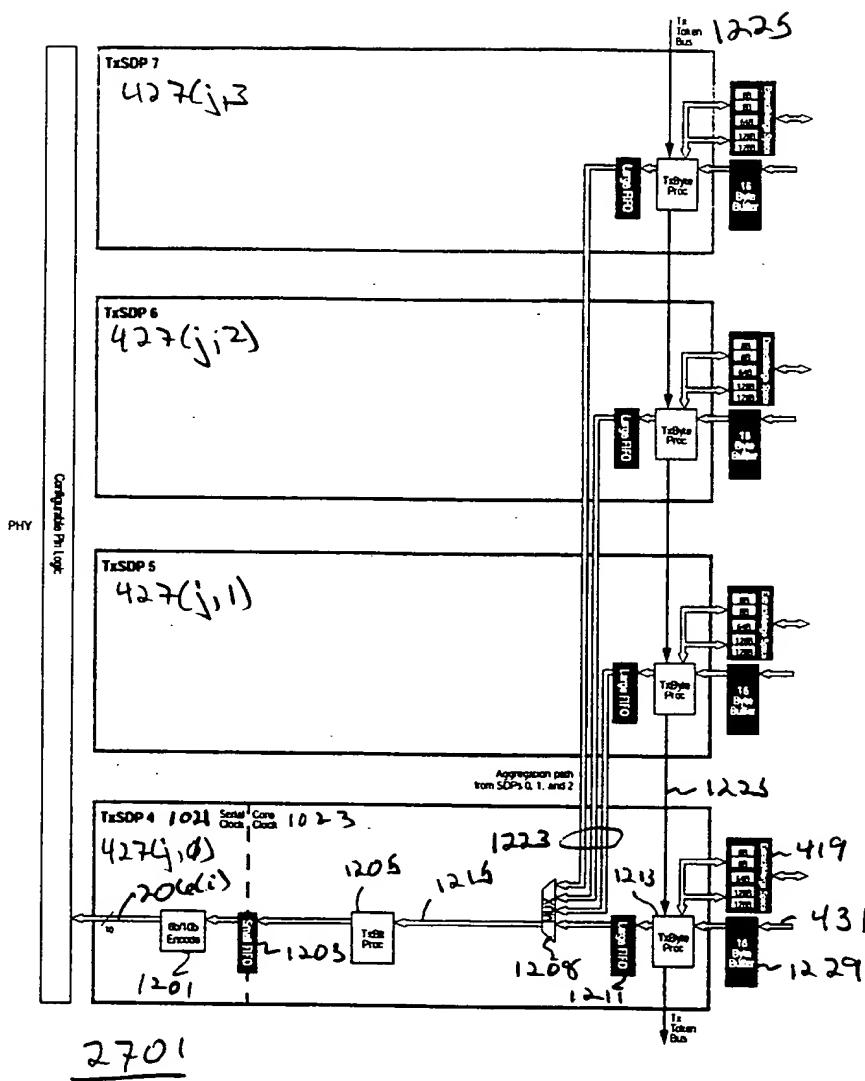


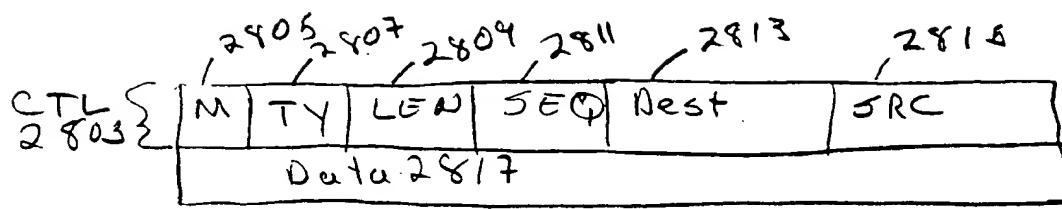
Fig. 27

09/674864

PCT/US99/10002

WO 99/59078

28 / 47



Rungbus message 2801

09/674864

0962334 0962000

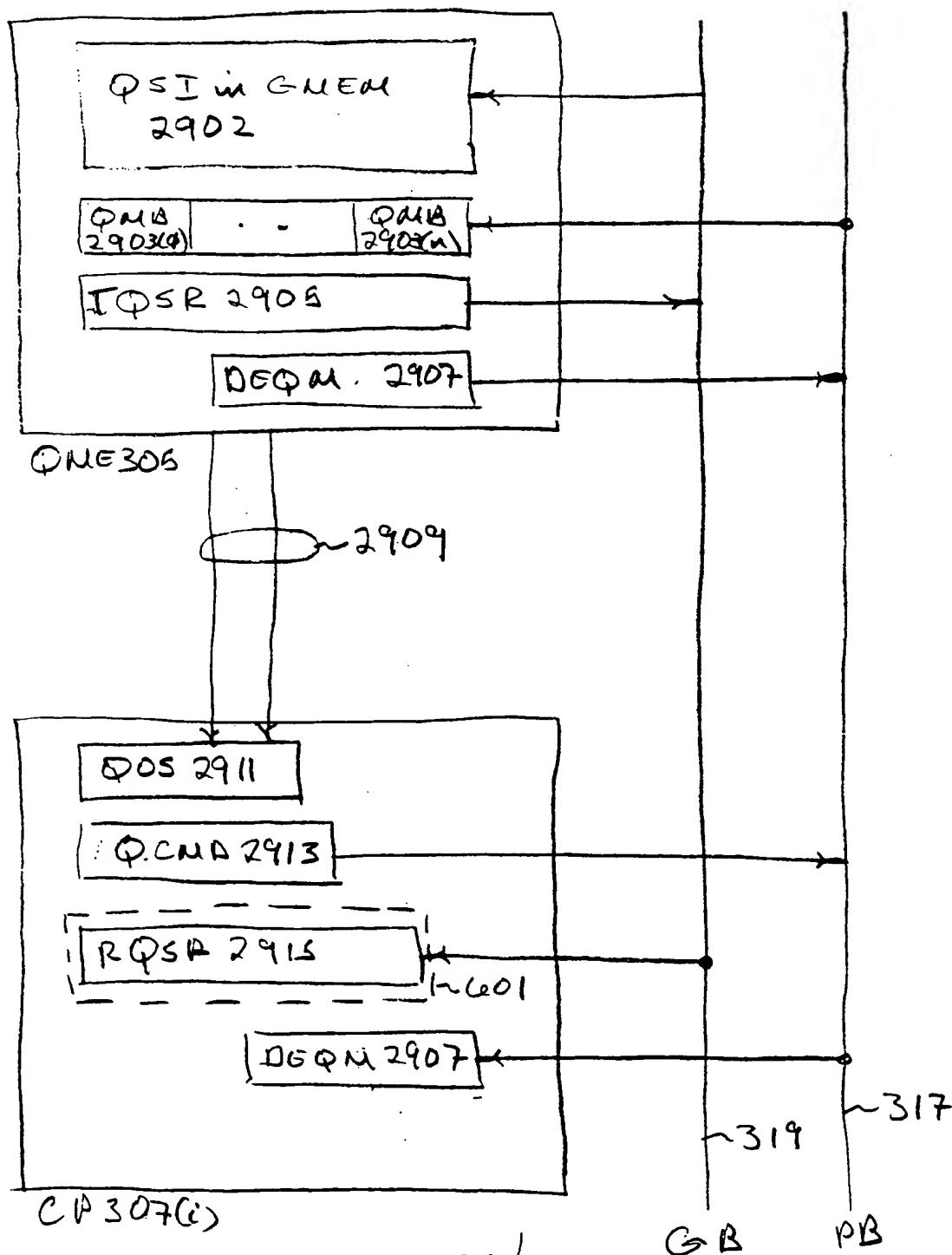


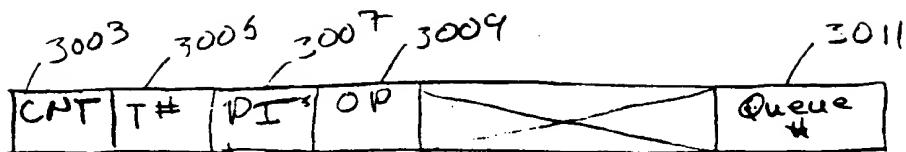
Fig. 29

09/674864

PCT/US99/10002

WO 99/59078

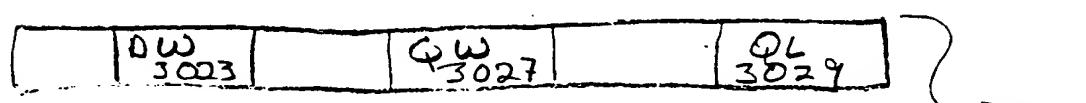
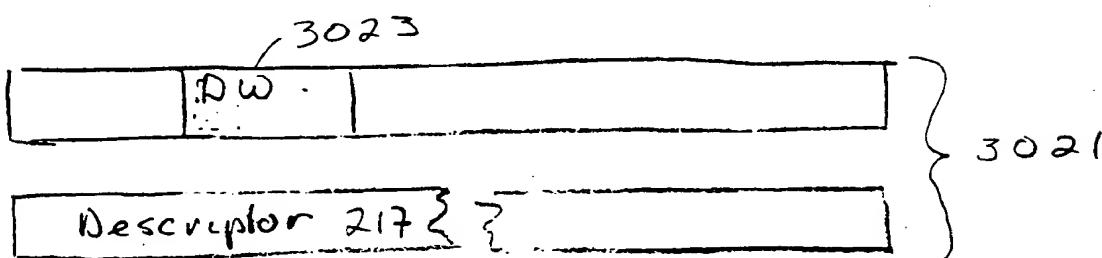
30 / 47



Queue Inst. Addr. 3001



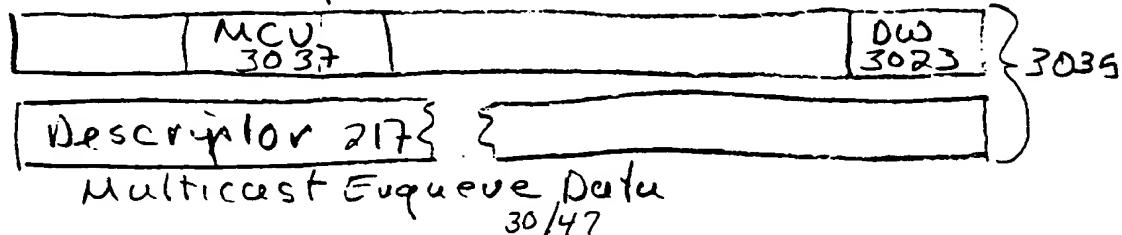
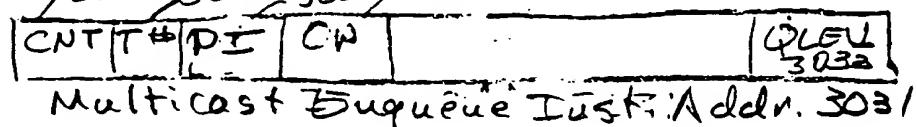
Configure Queue Write Data 3013



Descriptor 217

Unicast Enqueue Data

3003 3005 3007 3009



2913

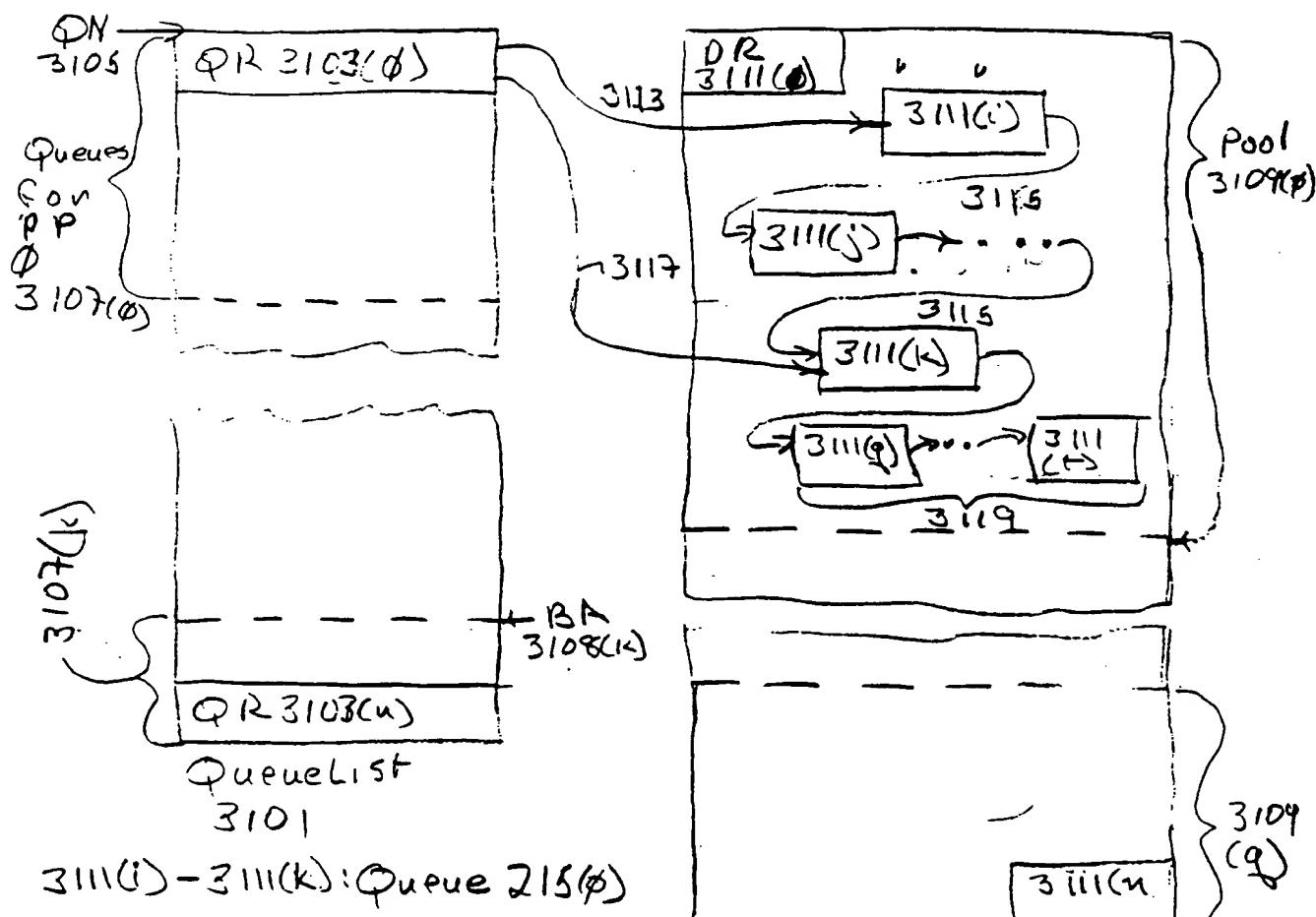
Fig. 30

09/674864

PCT/US99/10002

WO 99/59078

31 / 47



HD PTR 3113
TPTR 3117
QL 3129
TDW 3131
ADA 3133
QLL 3135
<u>3103</u>

DW 3137	IUC 3139
BT 233	
NDTR 3115	
<u>3111</u>	

Descriptor pools 3107

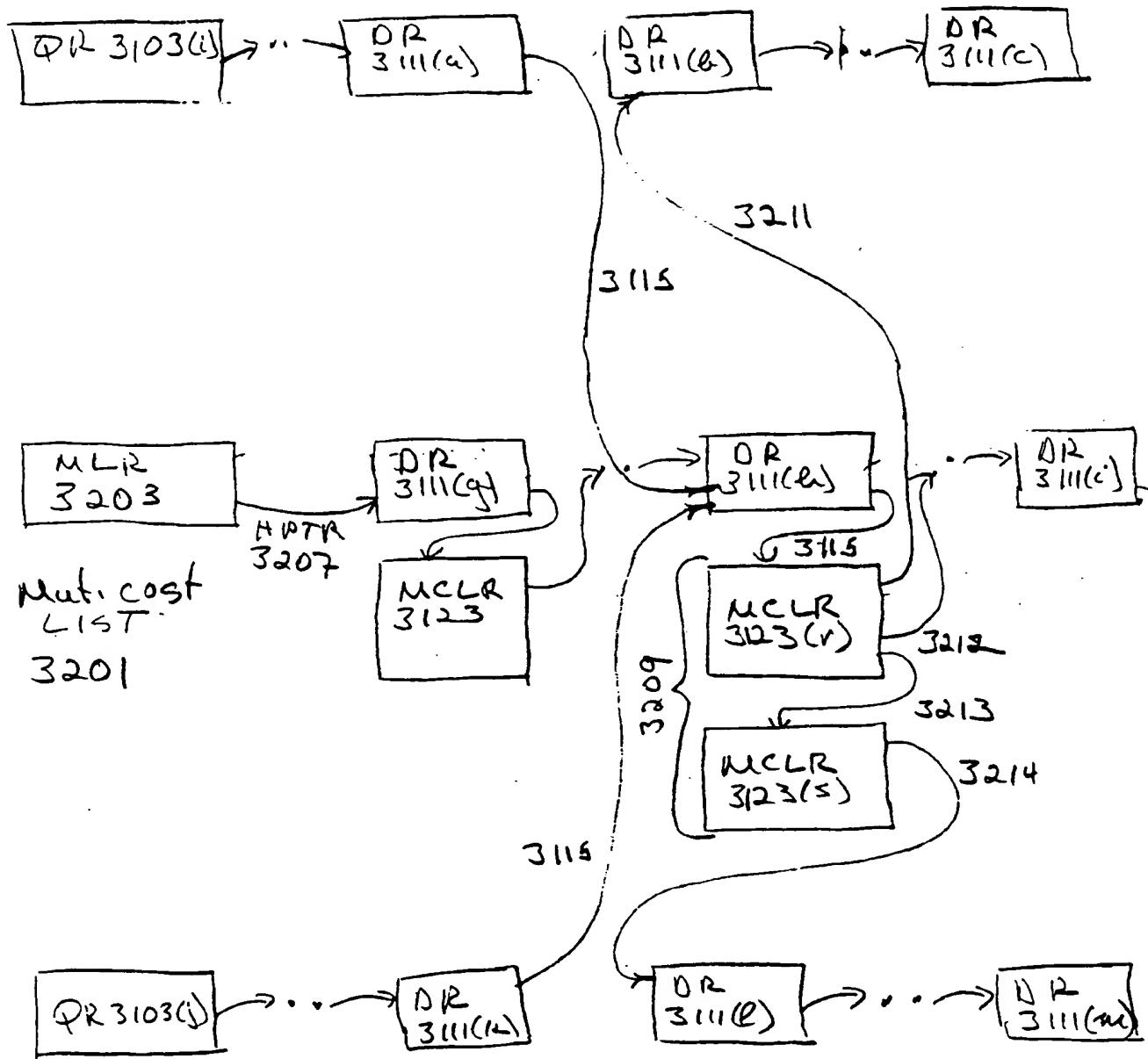
MCP 3123(φ)
3123(u)

MCRP 001 3121

Fig. 31

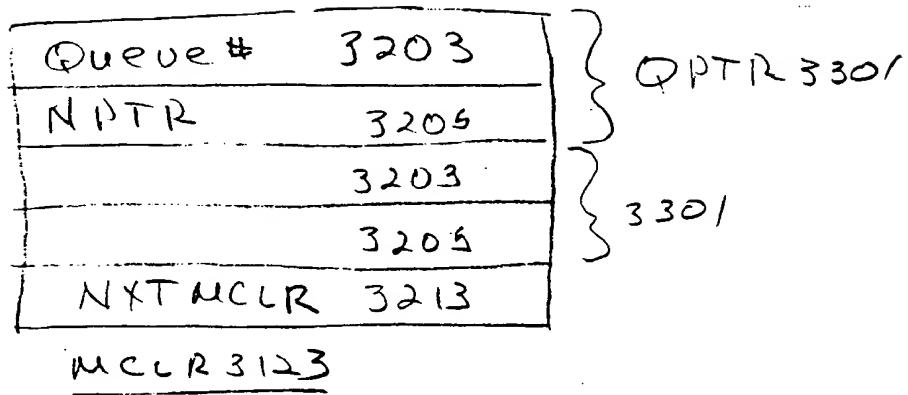
09/674864

## Unicast Queue 215(i)



## Unicast Queue 215(j)

09/674864



User #, Queueing Level # 3309

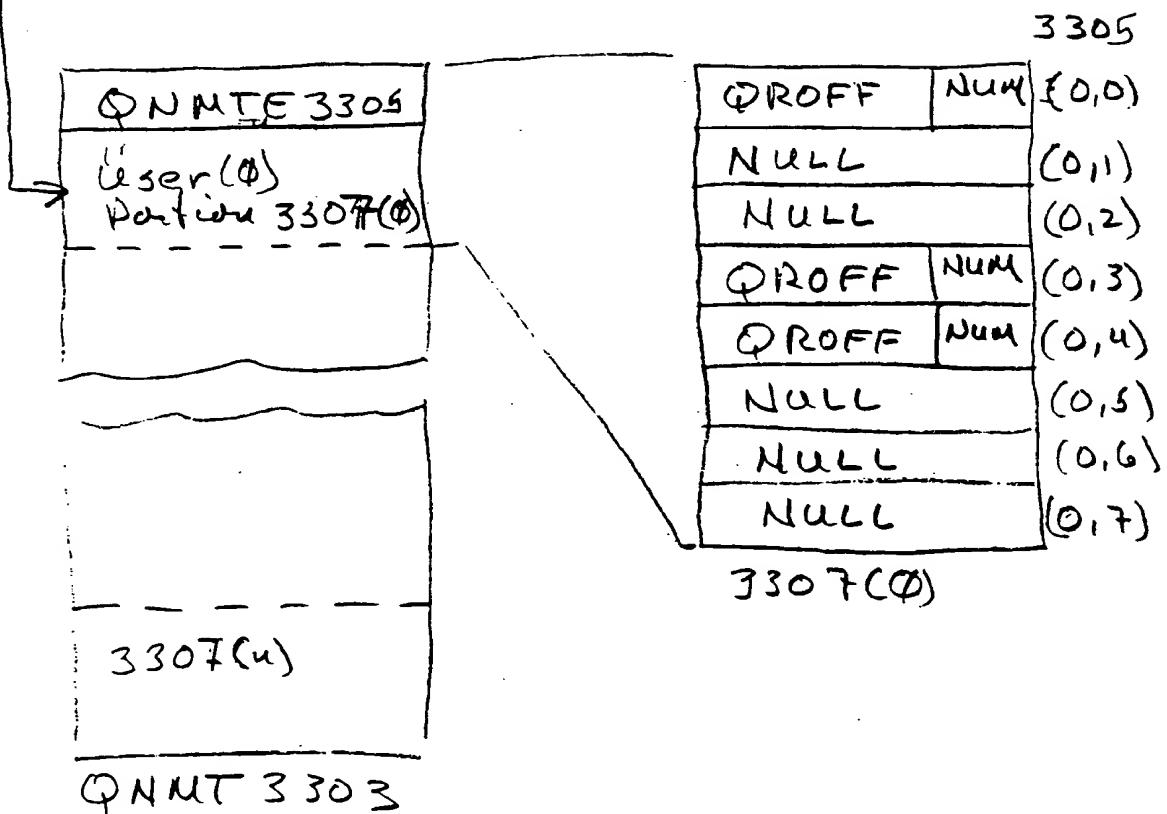


Fig. 33

09/674864

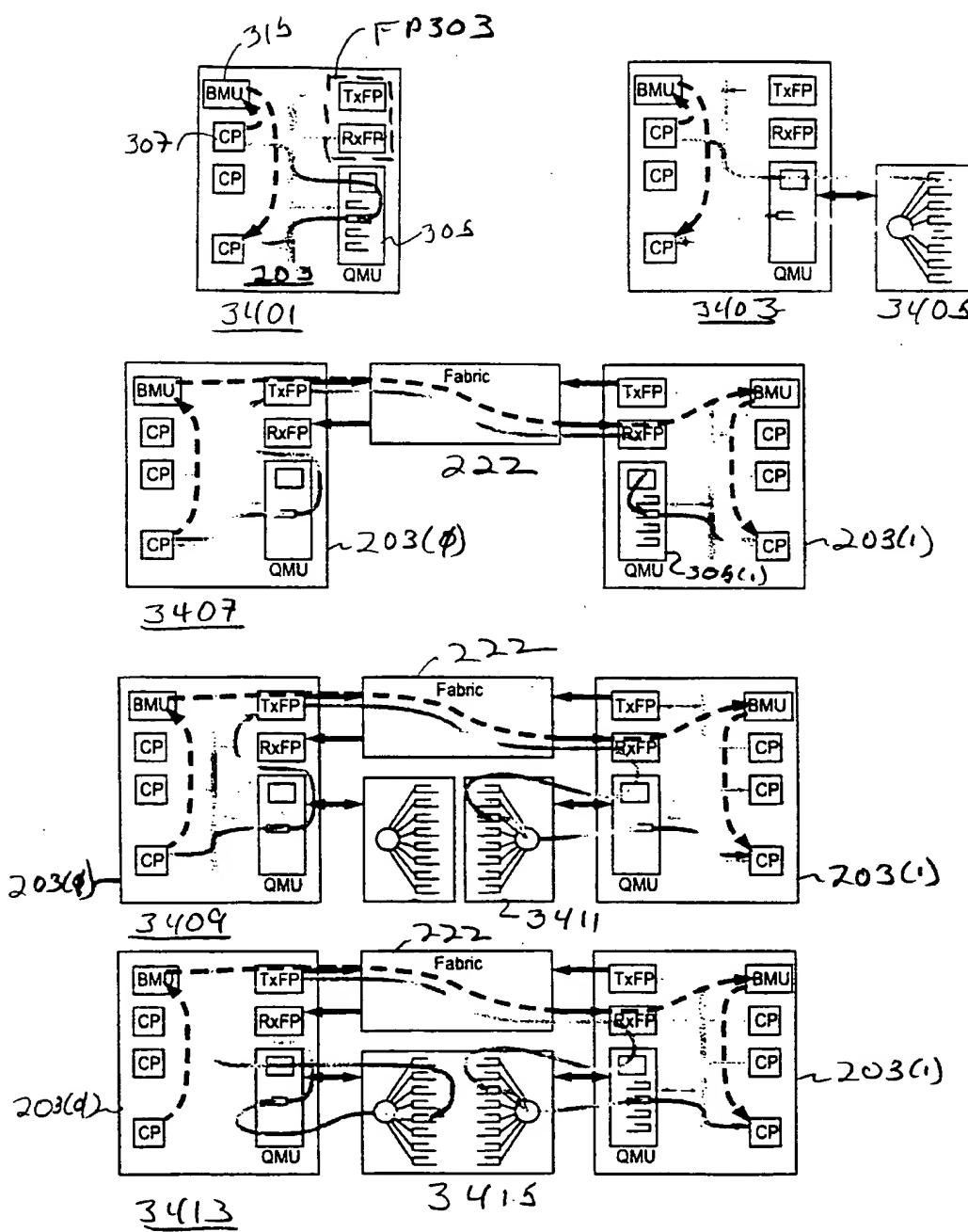
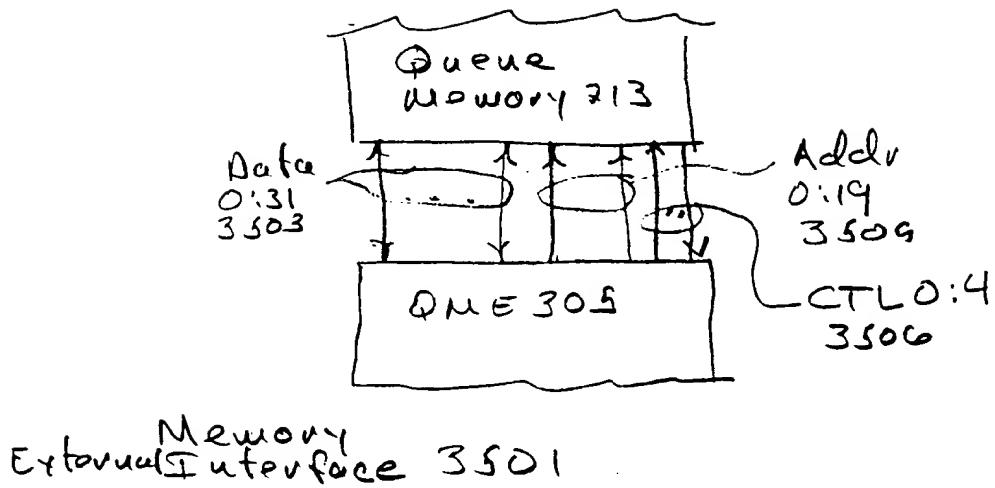
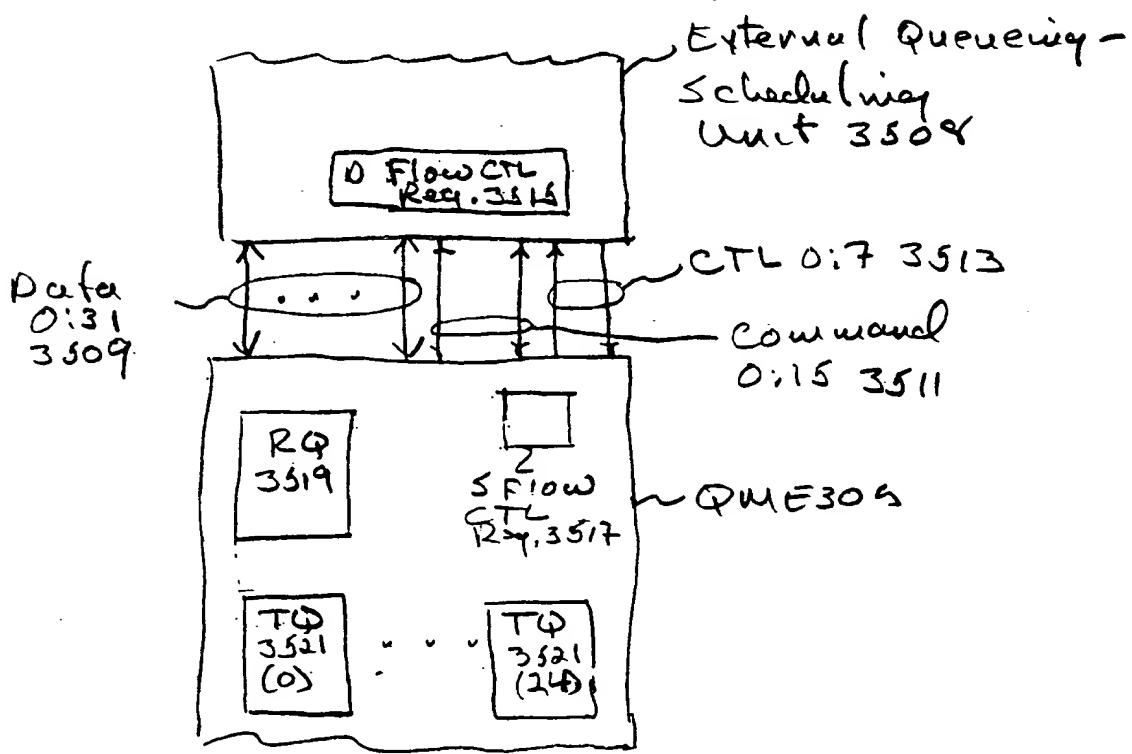


Fig. 34

09674864



External Memory Interface 3501

Scheduler External Interface 3507  
35/47

09674864

pins 3601  
Direction 3603

<u>3605</u> — Clk	1 DCP → SCHED
<u>3607</u> — D_Flow_Ctrl	3 DCP → SCHED
<u>3607</u> — S_Flow_Ctrl	1 DCP ← SCHED ; If = 0, the Scheduler can ; accept at least one descriptor.
<u>3611</u> — Xfer_Rqst	1 DCP ← SCHED ; If = 1, the Scheduler has at ; least one descriptor to transfer.
<u>3613</u> — Xfer_Ctrl	2 DCP → SCHED
Command Data	16 DCP <→ SCHED
Cmd_Parity	1 DCP <→ SCHED
<u>3509</u>	Descript Data 32 DCP <→ SCHED Data_Parity 1 DCP <→ SCHED
---	
Total	58

3507

First Command Code 3615

1	1	1	1	1	1	1	1	1	1	1
5	4	3	2	1	0	9	8	7	6	5

Destination DCP Processor Number 3617

3514

Fig. 36

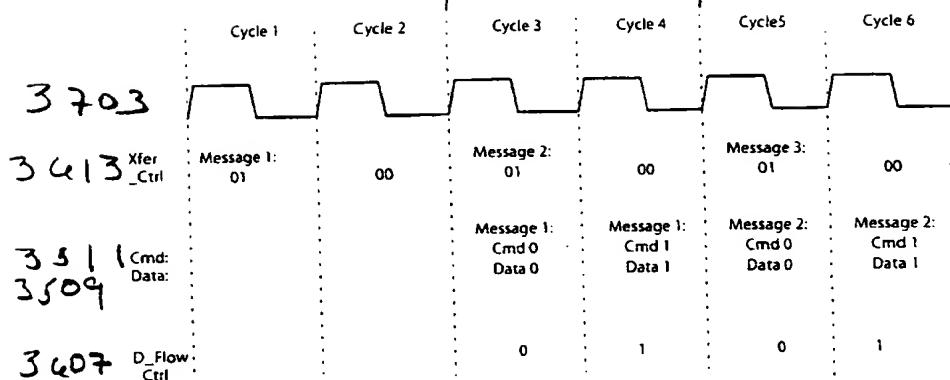
09/674864

PCT/US99/10002

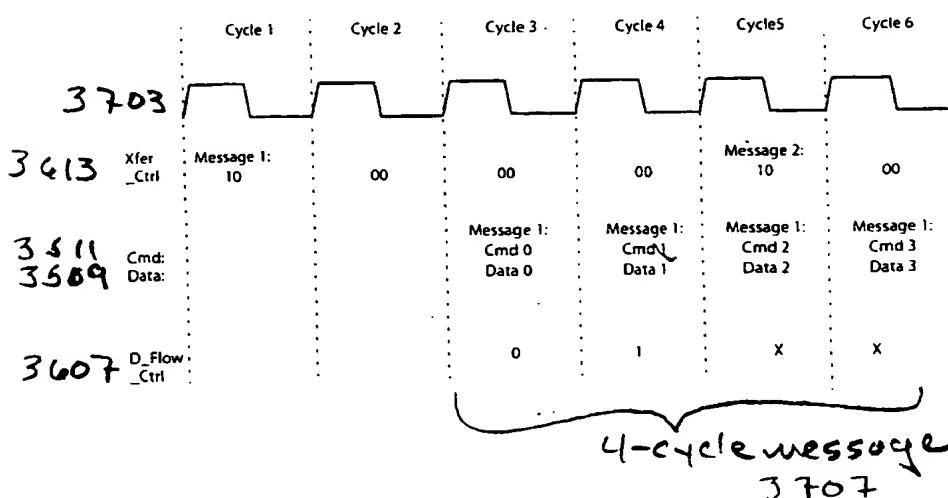
WO 99/59078

37 / 47

09/674864 PCT/US99/10002

2-cycle  
message 3702

3701: 2 2-cycle messages



3705: 2 4-cycle messages

Fig. 37

09/674864

PCT/US99/10002

WO 99/59078

38 / 47

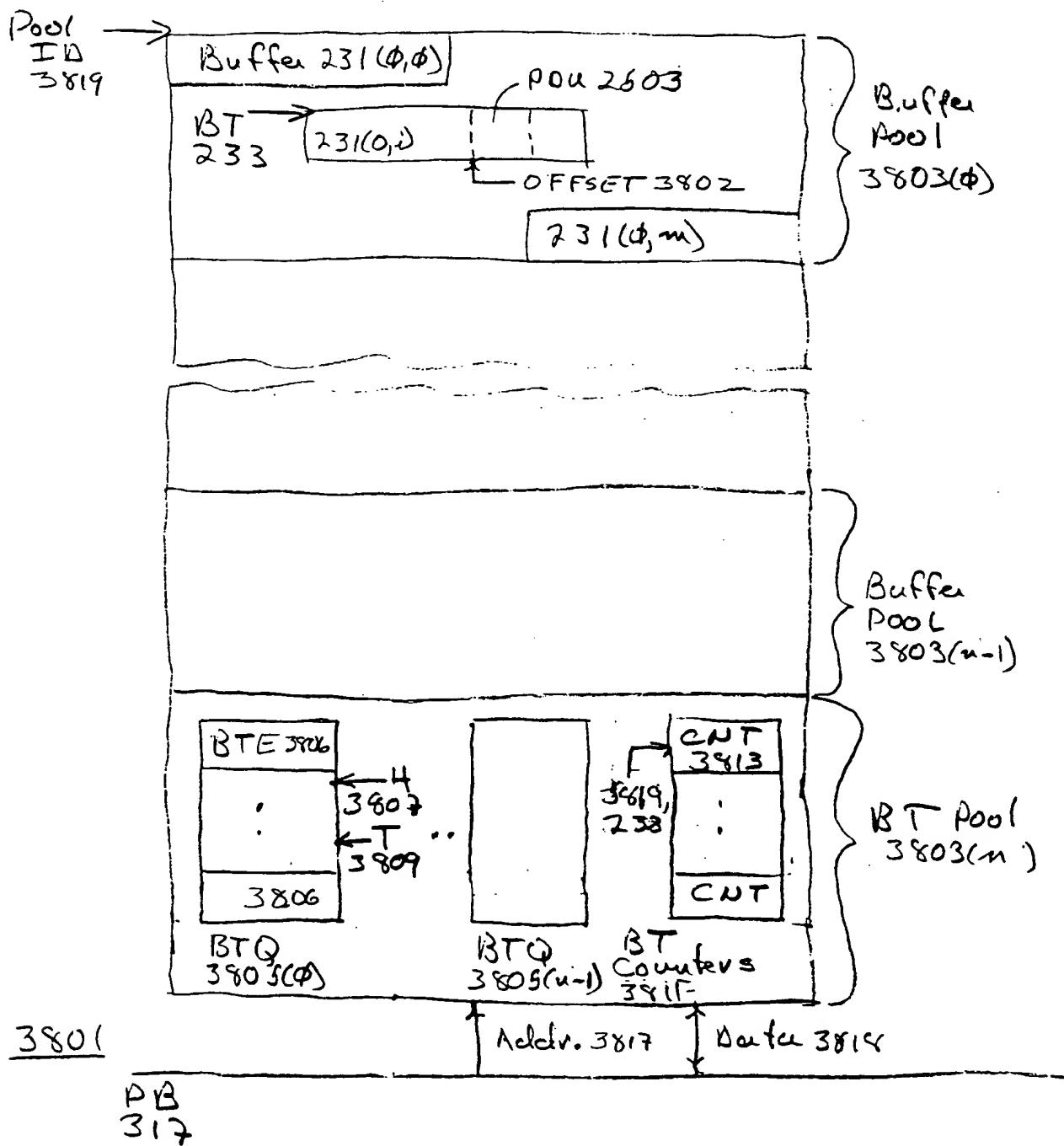


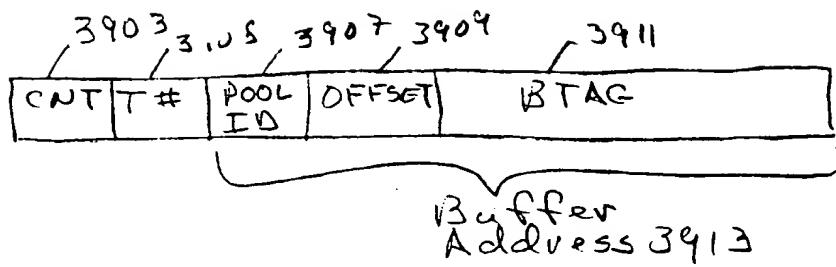
Fig. 38

09/674864

PCT/US99/10002

WO 99/59078

39 / 47



Payload bus

Buffer read/write command 3901

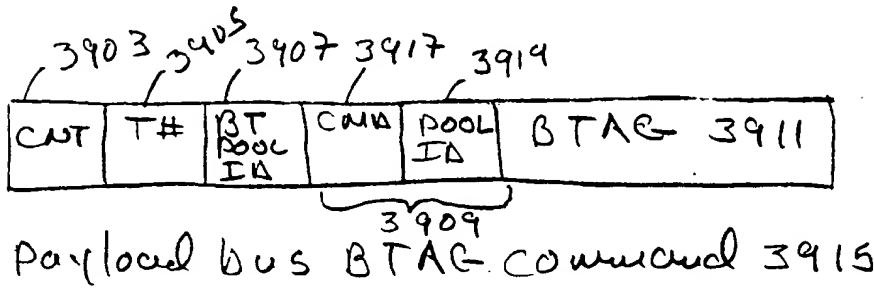


Fig. 39

09674364 - 0333001

09/67484

09/67484 033004

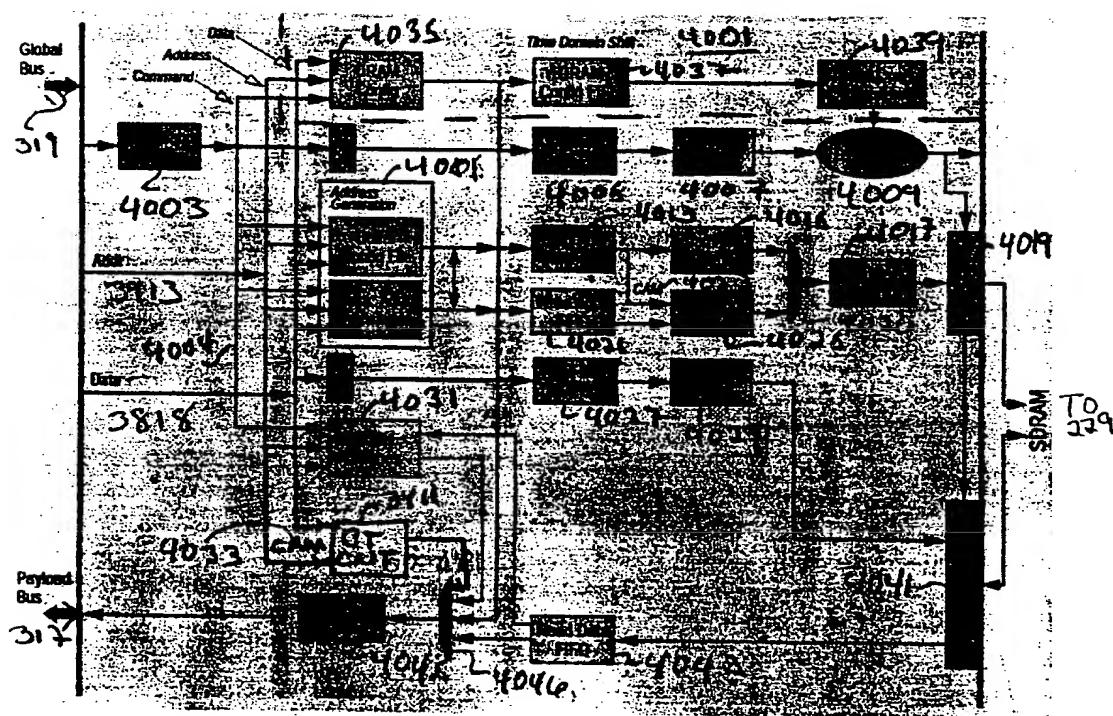
316

Fig 410

09/674864

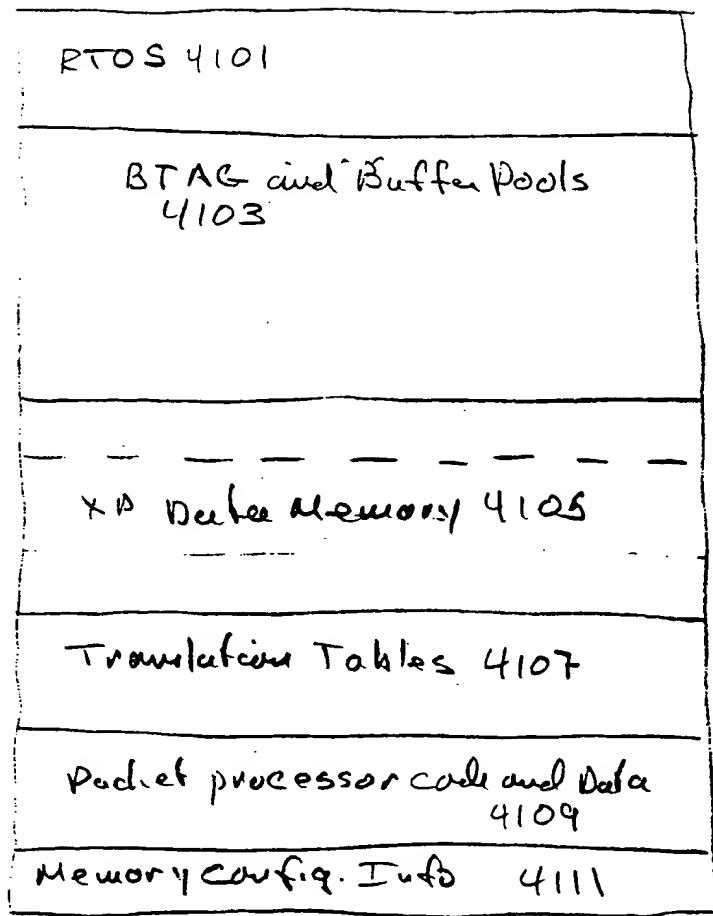
229

Fig. 411

09/674864

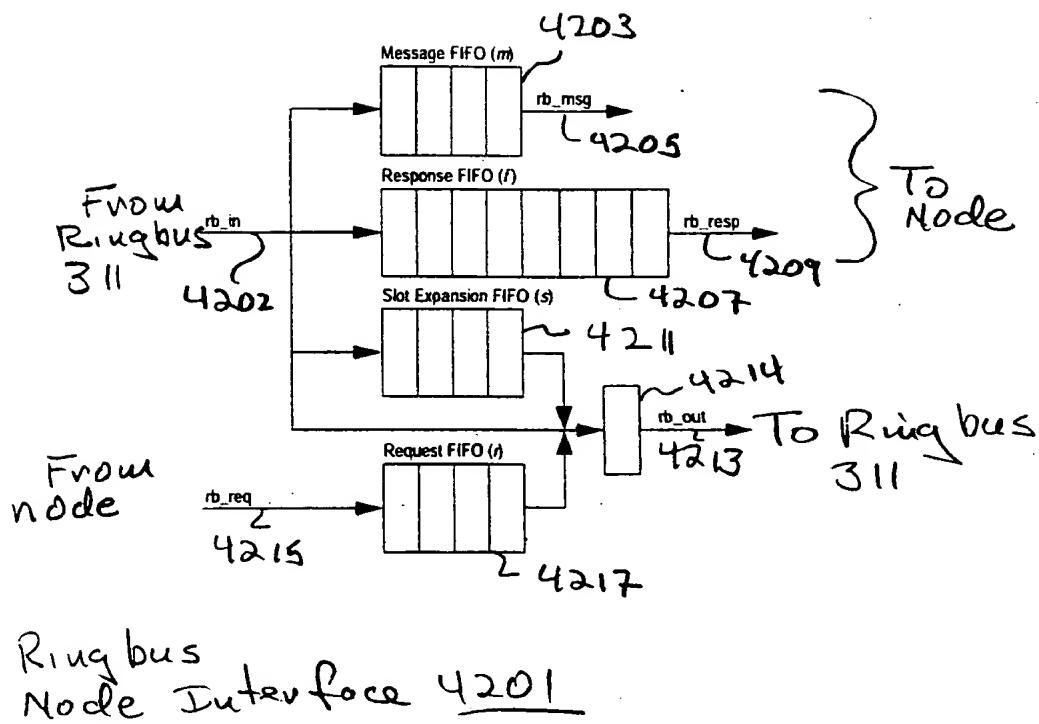
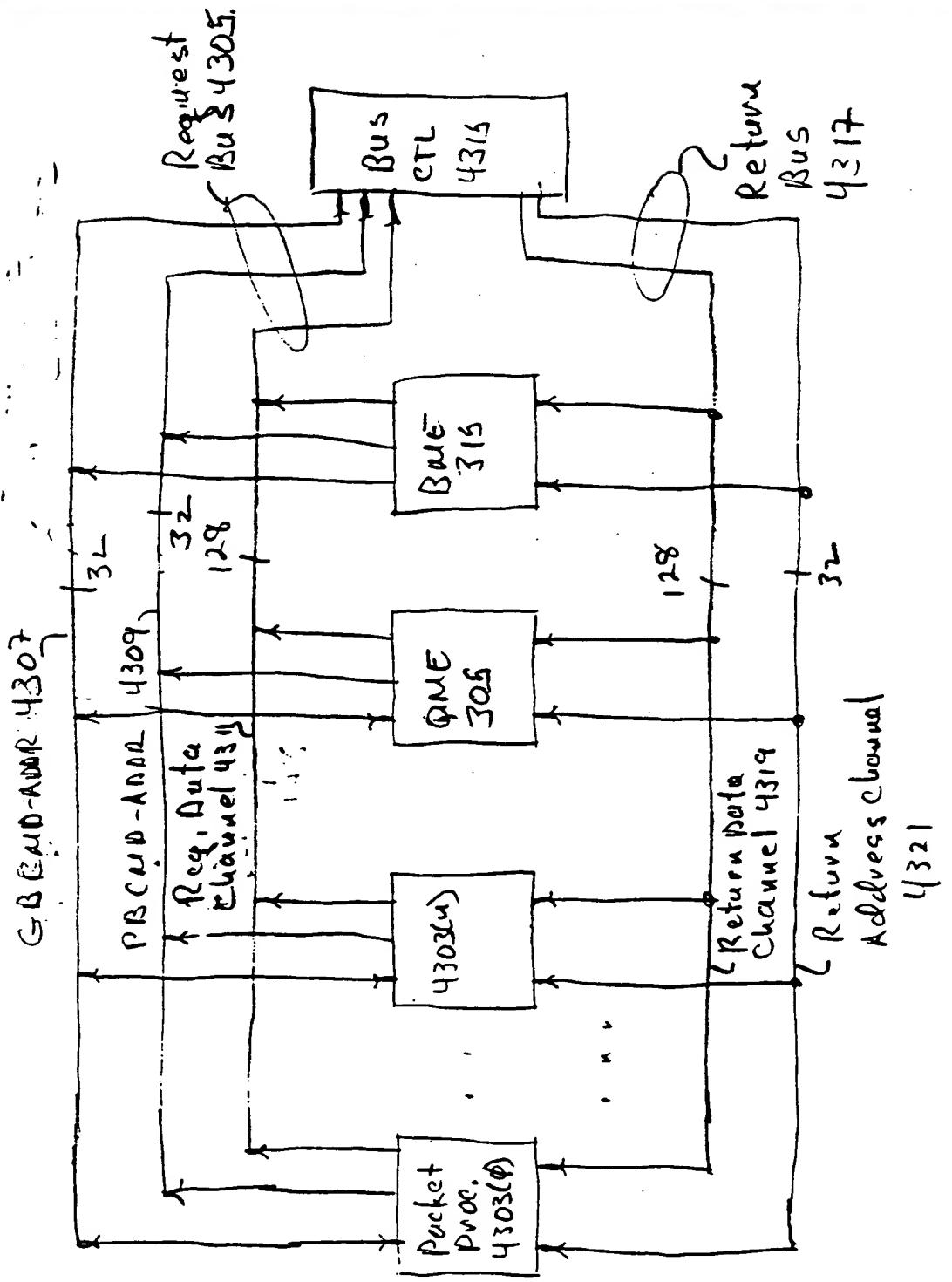


Fig. 42

09/07/1804



43/47

301

Fig. 4(3)

09/674864

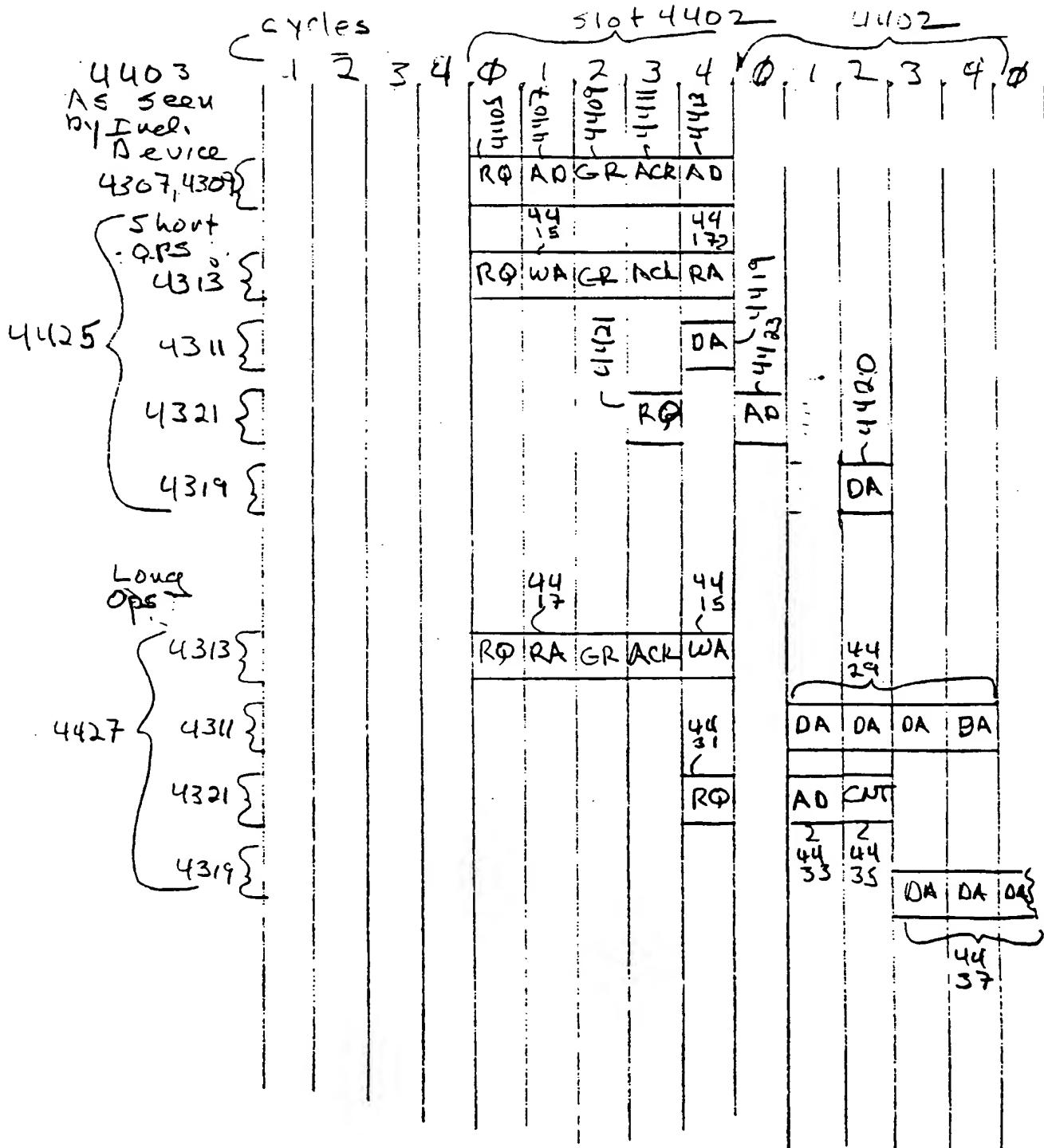
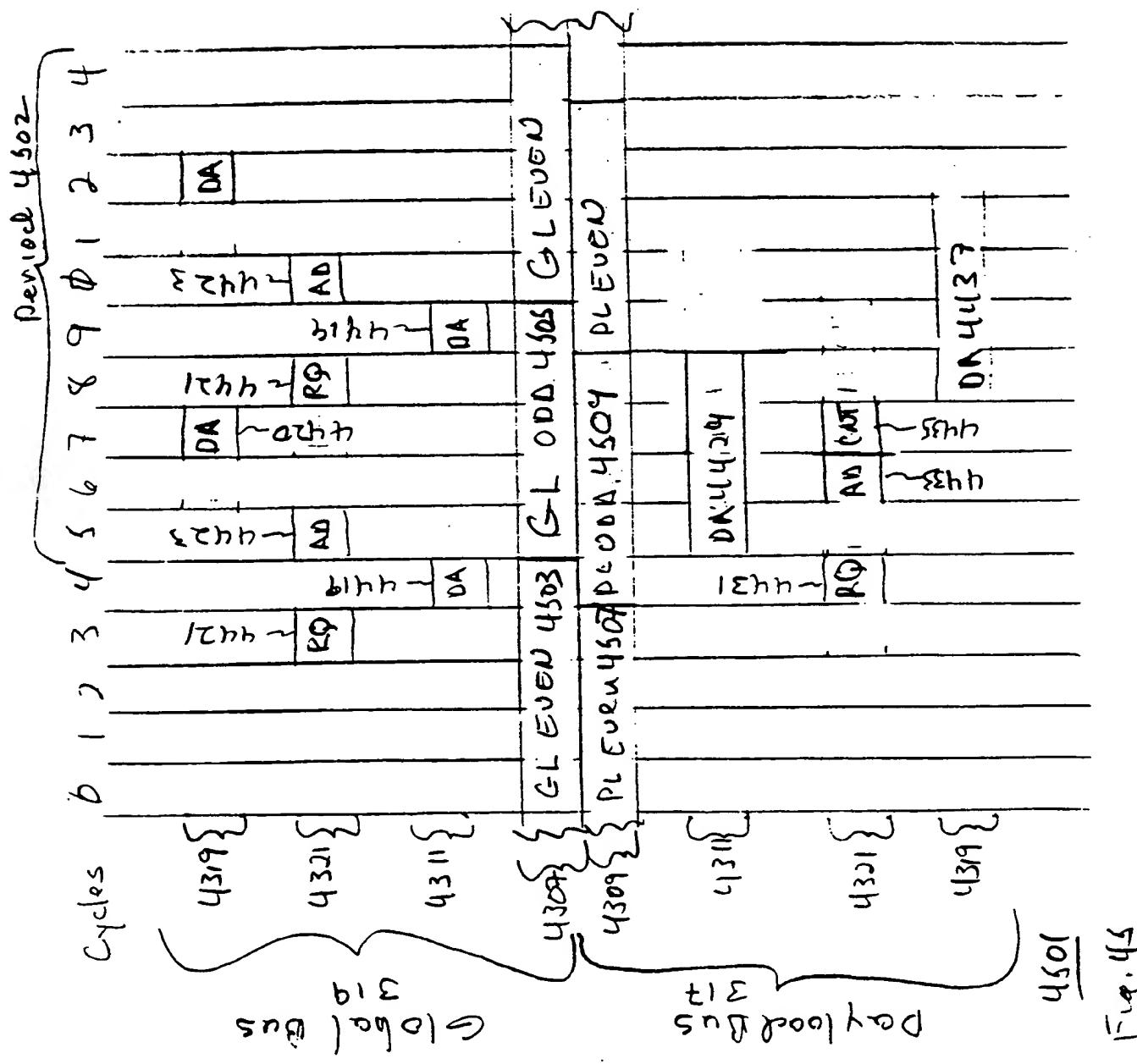
4401

Fig. 44

09/674864



09/07/98

4603 4605 4607 4609 4611 4613 4615 4617 4619

Pin	Purpose	RMII	OC-3	DS1	DS3	GMII (Tx)	GMII (Rx)	TBI (Tx)	TBI (Rx)	OC-12
CPO_0	outclk	REF_CLK	RCLK_H	TCLK	TCLK	nc	TCLK	nc	TCLK	
_1	inclk	CRS_DV	RCLK_L	RCLK	RCLK	CRS	nc	nc	nc	TCLKT
_2	data	TXD[0]	TXD_H	TDATA	TDATA	TXD[0]	nc	TXD[0]	nc	TXD[0]
_3	data	TXD[1]	TXD_L	TxFrame	TxFrame	TXD[1]	nc	TXD[1]	nc	TXD[1]
_4	data	RXD[0]	RXD_H	RDATA	RDATA	TXD[2]	nc	TXD[2]	nc	TXD[2]
_5	data	RXD[1]	RXD_L	RxFrame	RxFrame	TXD[3]	nc	TXD[3]	nc	TXD[3]
_6	data	TX_EN	SIGNAL_DET			TX_EN	nc	TXD[8]	nc	
CP1_0	outclk	REF_CLK	RCLK_H	TCLK	TCLK					
_1	inclk	CRS_DV	RCLK_L	RCLK	RCLK	COL	nc			
_2	data	TXD[0]	TXD_H	TDATA	TDATA	TXD[4]	nc	TXD[4]	nc	TXD[4]
_3	data	TXD[1]	TXD_L	TxFrame	TxFrame	TXD[5]	nc	TXD[5]	nc	TXD[5]
_4	data	RXD[0]	RXD_H	RDATA	RDATA	TXD[6]	nc	TXD[6]	nc	TXD[6]
_5	data	RXD[1]	RXD_L	RxFrame	RxFrame	TXD[7]	nc	TXD[7]	nc	TXD[7]
_6	data	TX_EN	SIGNAL_DET			TX_ER	nc	TXD[9]	nc	
CP2_0	outclk	REF_CLK	RCLK_H	TCLK	TCLK					
_1	inclk	CRS_DV	RCLK_L	RCLK	RCLK	nc	RCLK	nc	RCLK	RCLK
_2	data	TXD[0]	TXD_H	TDATA	TDATA	TXD[0]	nc	RXD[0]	RXD[0]	
_3	data	TXD[1]	TXD_L	TxFrame	TxFrame	TXD[1]	nc	RXD[1]	RXD[1]	
_4	data	RXD[0]	RXD_H	RDATA	RDATA	TXD[2]	nc	RXD[2]	RXD[2]	
_5	data	RXD[1]	RXD_L	RxFrame	RxFrame	TXD[3]	nc	RXD[3]	RXD[3]	
_6	data	TX_EN	SIGNAL_DET			TX_DV		RXD[8]	FP	
CP3_0	outclk	REF_CLK	RCLK_H	TCLK	TCLK					
_1	inclk	CRS_DV	RCLK_L	RCLK	RCLK		nc	RCLKN		
_2	data	TXD[0]	TXD_H	TDATA	TDATA	TXD[4]	nc	RXD[4]	RXD[4]	
_3	data	TXD[1]	TXD_L	TxFrame	TxFrame	TXD[5]	nc	RXD[5]	RXD[5]	
_4	data	RXD[0]	RXD_H	RDATA	RDATA	TXD[6]	nc	RXD[6]	RXD[6]	
_5	data	RXD[1]	RXD_L	RxFrame	RxFrame	TXD[7]	nc	RXD[7]	RXD[7]	
_6	data	TX_EN	SIGNAL_DET			RX_ER	nc	RXD[9]	LOCKDET	

Fig. 46 4601

09/674864

47 47

4703	4706	4707	4709	4710	4711
Date Cutting	Recell- ment max	Rego- tional max	Regio- nally max	Regio- nally max	Regio- nally max

4701

4715	4719	4723	4729	4731	4735
<input checked="" type="checkbox"/>					
C1	C2	C3	C4	C5	C6
<input checked="" type="checkbox"/>					
R1	R2	R3	R4	R5	R6

4713    4717    4721    4725    4731

Fig. 417